456-TP-004-001

ECS Project Training Material Volume 4: Science Software Integration & Test

Technical Paper

May 1997

Prepared Under Contract NAS5-60000

RESPONSIBLE ENGINEER

Kenneth Prickett, Project Engineer — Date
EOSDIS Core System Project

SUBMITTED BY

Tom Hickey, M&O Deputy Manager — Date
EOSDIS Core System Project

Hughes Information Technology Systems
Upper Marlboro, Maryland

This page intentionally left blank.

Abstract

This is Volume 4 of a series of 10 volumes containing training material for the Pre-Release B Testbed of the Earth Observing System Data and Information System (EOSDIS) Core System (ECS). This lesson provides a detailed description of the process required to install integrate science software PGEs into the EOSDIS environment, and once installed, test the new science software to verify its operability.

***Keywords**:* training, instructional, design, course, objective

This page intentionally left blank.

Contents

Introduction

Science Software Integration and Test (SSI&T) Overview

Science Software Integration and Test (SSI&T) Preparation and Setup

Acquiring and Unpacking Delivered Algorithm Package

Science Software Configuration Management

Preparation of Earth Science Data Types

Science Software Integration and Test (SSIT) Manager

Standards Checking of Science Software

Compiling and Linking Science Software

Running a PGE in a Simulated SCF Environment

Examining PGE-Produced Log Files

File Comparison

Data Visualization

Updating the PDPS Database and IMF Data Server

PGE Planning and Processing

Post-Processing Activities

Troubleshooting and General Investigation

Practical Exercise

Slide Presentation

This page intentionally left blank.

Introduction

Identification

Training Material Volume 4 is part of a series of Technical Papers that will be used to teach M&O concepts to the M&O staff at the LaRC, NSIDC and EDC DAACs.

Scope

Training Material Volume 4 defines the tasks required to perform Science Software Integration & Test (SSI&T). This lesson is designed to provide the operations staff with sufficient knowledge and information to satisfy all lesson objectives.

This document reflects the August 23, 1995 Technical Baseline maintained by the contractor Configuration Control Board (CCB) in accordance with ECS technical direction #11, dated December 6, 1994.

Purpose

The purpose of this Technical Paper provides a detailed course of instruction that forms the basis for understanding SSI&T. Lesson objectives are developed and will be used to guide the flow of instruction for this lesson. The lesson objectives will serve as the basis for verifying that all lesson topics are contained within this Student Guide and slide presentation material.

Organization

This document is organized as follows:

Introduction:	The Introduction presents the document identification, scope, purpose, and organization.
Student Guide:	The Student Guide identifies the core elements of this lesson. All Lesson Objectives and associated topics are included.
Slide Presentation:	Slide Presentation is reserved for all slides used by the instructor during the presentation of this lesson.

This page intentionally left blank.

Science Software Integration and Test (SSI&T) Overview

Lesson Overview

This lesson will provide you with the complete process by which a new science algorithm from the science community is received, integrated with ECS, tested and distributed. While the basic concept for accomplishing these tasks is fairly straight forward, many staff members are utilized in order to accomplish the process.

Lesson Objectives

Overall Objective - The objective of the SSI&T lesson is to provide a detailed description of the process required to install and test PGEs in the EOSDIS environment.

Condition - The student will be given an operational system with supporting equipment, the Student Guide.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 1 - The student will acquire the delivered algorithm package by using FTP.

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 2 - The student will use ClearCase™ to create a view, create a new directory , import files and check in and out files.

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 3 - The student will be able to prepare Earth Science Data Types (ESDTs) for use on ECS.

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 4 - The student will perform routine running of the SSIT Manager Graphical User Interface(GUI)

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 5- The student will verify that FORTRAN 77, FORTRAN 90, C and Ada source files are compliant with the ESDIS Data Production Software and Science Computing Facility (SCF) standards and guidelines document and will search source files for prohibited functions.

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 6- The student will compile Product Generation Executables (PGEs) and link SCF and DAAC version of Science Data Processing (SDP) toolkits and update files by configuring the SDP Toolkit environment, updating the process control file and compiling status message facility files.

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 7 - The student will update the PDPS Database and data server by inserting operational metadata, science metadata, science software archive data, static files and executables into the data server.

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 8 - The student will run a PGE executable and provide resource requirements for a PGE.

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 9 - The student will perform HDF, ASCII, and binary file comparison using the GUI and hdiff.

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Specific Objective 10 - The student will perform problem tracking on science software by verifying that the range values for the metadata control files are valid, reviewing the Wabi directory, examining the SDP Toolkit status messages, examining the MSS log files and extracting prologs from the source files.

Condition - The student will be given an operational system with supporting equipment.

Standard - The student will use the tools in accordance with prescribed methods and complete required procedures without error.

Importance

The Science Software Integration and Test lesson will provide a review of the process that allows the SSIT team, Production Planner and Production Monitor to retrieve, install, check, compile, test and monitor science software.

This page intentionally left blank.

Science Software Integration and Test (SSI&T) Preparation and Setup

Key Operator Roles

Science Coordinator: Provide support to Instrument Teams for the development and integration of science software into the DAAC ECS system. Perform standard checking on all delivered software including source code, scripts, process control files and related documentation.

Science Data Specialist: Serves as a point-of-contact for planning, integrating, testing, and operating science software.

CM Administrator: Record, report, manage and distribute new and updated science software.

Science Software I & T Support Engineer: Provide support to Instrument Teams for the development, integration, test and problem resolution of science software.

Production Planner: Populate, maintain and schedule the production planning database for science software.

COTS Software Tools

ClearCase™: This tool is used as the ECS software configuration management tool. ClearCase™ provides a mountable file system which is used to store version-controlled data, such as source files, binary files, object libraries and spreadsheets.

Distributed Defect Tracking System (DDTS): This tool is used to electronically process configuration change requests (CCRs). DDTS will prompt the user for relevant information, identify the request and will mail these requests to pre-designated personnel.

General Process

The SSI&T process consist of two phases:

- Pre-SSI&T phase - During this phase the Delivered Algorithm Package (DAP) is inspected, and tested in a non-production environment.
- Formal SSI&T phase - During this phase, the Product Generation Executives (PGEs) are integrated with the DAAC version of the SDP Toolkit and executed on the ECS PDPS platform.

- **Key Terms:**
  - **Product Generation Executives (PGEs)** - The smallest scheduled unit of science software.
  - **Delivered Algorithm Package (DAP)** - An ensemble of PGE source code, makefile, documentation, and other related files delivered in a package from the SCF to the DAAC.
  - **Process Control File (PCF)** - Relate logical identifiers to physical files and other parameters required by the PGE.
  - **Strings** - The processing hardware on which the science software runs.

The DAPs will eventually be integrated onto the PDPS and be used to produce the output data as determined by the algorithms and eventually be provided to the subscribing user. Before the PGE is integrated into a production environment, extensive testing on the software must be performed.

The following list provides a suggested, logical "road map" for getting science software tested and integrated into the ECS system. This list is not intended to cover every situation and variations may be required.

Pre SSI&T Phase

1. Acquire the DAP - This is performed by electronic (FTP) transfer or by hard media.
2. Inspect the DAP - The DAP contents are checked by the Science Data Specialist to verify that the contents match the packing list, agreed-upon directory structures are employed, location of files are correct, and all intended files and directories are present.
3. The Science Data Specialist reviews the delivered documentation.
4. The Science Data Specialist requests that the CM Administrator place the DAP under Configuration Management control using ClearCase™.

5. The SSI&T team checks the science software for standards compliance using the Process Control File Checker to check process control files, and the Prohibited Function Checker to check source files.

6. The SSI&T team builds the science software into PGEs using the SCF version of the SDP Toolkit. Compile all source code. Link object code with appropriate libraries.

7. The SSI&T team runs the PGEs from the UNIX command line on the SGI.

8. The SSI&T team collects performance statistics for the PGEs.

9. The SSI&T team examines the output log files from the PGE runs for any anomalous messages.

10. The SSI&T team compares the output product data with the delivered test data using the file comparison tools.

11. The SSI&T team reports any science software problems using the DDTS NCR process.

12. The SSI&T team reports any ECS problems using the DDTS NCR process.

13. The SSI&T team collects and logs all lessons learned.

Formal SSI&T Phase

1. Register ESDTs on the Science Data Server. Registration links the PGE to all input and output ESDTs which allows the PGE to run within the PDPS.

2. Register the PGEs in the PDPS database.

3. Build the science software into PGEs with the DAAC version of the SDP Toolkit.

4. Using the Production Request Editor, submit an individual Production Request that results in the PGE being run once.

5. Examine the output log files from the PGE runs for any anomalous messages.

6. Compare the output product data with the delivered test data using the file comparison tools.

7. Using the Planning subsystem, initiate more complex Production Requests.

8. Acquire and examine the output products generated by the PGEs.

9. Using electronic or hard media transfer methods, distribute the data products to the Instrument Teams for their review.

Starting the SSIT Manager

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

- If you access the workstation through a remote login (rlogin), you must enter **xhost +** and then press the **Enter** key.
- Once you have entered xhost +, but prior to the remote login, enter **setenv DISPLAY** <local workstation IP address>**:0.0** where the local workstation IP address represents the IP address you where you are located**.**
- You may need to setup the terminal so that the remote host is displayed on your screen. (Sun machine) This is done by clicking on the **Application Manager** icon (the file drawer located at the bottom of the screen), followed by the **Desktop Tools** icon**,** followed by the **Terminal Consol** icon, then typing **xhost + aitn1sun** or **aitn2sun.gsfc.ecs.nasa.gov** in the display window.

3 Perform a remote login by typing **rlogin [aitn1sun or aitn2sun]** then press **Enter.**

- The **Enter Password** prompt is displayed.

4 Enter the **password** then press **Enter.**

5 Enter the directory where the setup script is located by typing **cd [**directory name**]** then press **Enter**.

6 Source the setup script by typing **source [**script name**]** then press **Enter.**

- The setup script contains the following directory paths and commands:

  cd /usr/ecs/Rel_A/CUSTOM/bin/DPS

  DpAtMgr ConfigFile /usr/ecs/Rel_A/cfg/DpAtMG_GSFC.CFG &

7 Enter the script alias by typing **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

Acquiring and Unpacking Delivered Algorithm Package

The SSIT team receives the science software from the science community. The following procedures will provide the SSIT team with the procedural steps that are used acquire files.

Acquiring the Algorithm Package via FTP

Acquiring the science software via FTP is one method that the SSIT team will use in order to receive the science software.

Acquiring the Algorithm Package via FTP

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At a UNIX prompt, type **cd *DeliveryPathname***, press **Enter.**

- The ***DeliveryPathname*** is the full path name to the directory that has been set aside for ftp pull of DAPs from the Instrument Team. For example, **cd /inbox/ASTER**, press **Enter.**

4 At a UNIX prompt, type **ftp *machineIPaddress***, press **Enter**.

- The ***machineIPaddress*** is the IP address or fully qualified domain name of the remote SCF machine. For example, **ftp 192.116.53.2**, press **Enter**. The remote machine will likely display some messages and then prompt for a login name.
- An ftp session is established.

5 At the ftp prompt on the remote machine, enter user login name, press **Enter**.

- The ftp prompt will look like **Name (*machinename*:*username*):** where ***machinename*** will be the name of the remote machine and ***username*** will be the user name of the user initiating the ftp connection. Enter a user login name which should be known. The remote machine will typically respond with **331 Password required for *username*:**

6 At the ftp prompt on the remote machine, enter user password, press **Enter**.

- The ftp prompt will look like **Password:** Enter the password which should be known. The remote machine will typically respond with **230 User *username* logged in** and display the **ftp>** prompt for further ftp commands.

7 At the ftp prompt on the remote machine, type **cd *DAPpathname***, press **Enter**.

- The ***DAPpathname*** is the full path name to the directory on the remote machine containing the DAP to retrieve. For example, **cd /pub/outbox/**, press **Enter**. The directory location should be known.

8 At the ftp prompt on the remote machine, type **binary**, press **Enter**.

- The **binary** command causes subsequent file transfers to be in binary mode, preserving the integrity of the file to retrieve without interpretation (as would be done in ASCII mode. The system will typically respond with the message **200 Type set to I** indicating that binary mode has been set.

9 At the ftp prompt on the remote machine, type **get *DAPfilename***, press **Enter**.

- The ***DAPfilename*** is the file name of the DAP to retrieve. For example, type **get ASTER_PGE3_V2.0_12151998.tar.Z**, press **Enter**. The user may need to type **dir** and then **Enter** to display a listing of the files in the current directory. The system will likely display several lines of messages once the transfer has completed. For large files, this may take a long time (minutes to hours depending upon the size of the DAP and the bandwidth of the connection).

10 At the ftp prompt on the remote machine, repeat step 9 or type **quit**, press **Enter**.

- Typing **quit** and pressing **Enter** closes the ftp connection with the remote machine.

- Retrieve other DAP files by repeating step 9. The DAPs retrieved will reside in ***DeliveryPathname*** on the local machine.

Unpacking a DAP

Once a DAP has been acquired via electronic means or physical media, it typically needs to be unpacked before its contents are accessible for SSI&T. Several mechanisms are available under standard UNIX for packing and unpacking files to and from a file archive, the most common being UNIX *tar*.

Unpacking a DAP

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At a UNIX prompt, type **cd *UnpackPathname***, press **Enter.**

- The ***UnpackPathname*** is the path name of the directory that has been set aside for unpacking of DAPs. For example, **cd /staging/ASTER**, press **Enter**.

4 If necessary, at a UNIX prompt, type **uncompress *PackedDAP.Z***, press **Enter**.

- The ***PackedDAP.Z*** is the file name of the compressed DAP file. For example, **uncompress ASTER_PGE3_V2.0_12151998.tar.Z**, press **Enter**. The file name extension of **.Z** is a convention indicating UNIX compressed files. The **uncompress** utility expects this file name extension by default. A resulting error may indicate that the DAP file was not compressed or that another compression utility was used. If the file name extension was **.Z**, the uncompressed version will have the same file name but without the **.Z**, for example ***PackedDAP***.

5 At the UNIX prompt, type **tar xvf *PackedDAP***, press **Enter**.

- The ***PackedDAP*** is the file name of the uncompressed DAP file. For example, **tar xvf ASTER_PGE3_V2.0_12151998.tar**, press **Enter**. The tar archive will be unpacked in the current directory. If the archive contained directories and subdirectories, these will be created by the tar utility and populated by the files that belong.

- These files are now ready to be CM controlled in ClearCase

This page intentionally left blank.

Science Software Configuration Management

The CM Administrator and System Administrator are key players in the SSI&T process. The CM Administrator receives the science software from the Science Data Specialist, places these files into a directory and request that the System Administrator place the files under configuration control by using the ClearCase™ tool. The science software is then tested by the SSI&T team and once the science software has successfully been tested, and upon direction from the CCB, the files are distributed to the Production Planner for placement on production server.

The CM and System Administrator need a good understanding of the ClearCase™ tool. ClearCase™ will be used to create a view, create a new directory, import files into the temporary subdirectories, and check-in and check-out files.

ClearCase™ Overview

The CM Administrator will be required to create a temporary SSI&T work area to review the files. Once reviewed, the System Administrator will place these files under configuration control. In order to accomplish this task, a view must be created in ClearCase™. A view is necessary in order to make visible and accessible files and directories that have been checked into a Versioned Object Base (VOB).

A Versioned Object Base is defined by the following characteristics:

- A mountable file system which stores version-controlled data, such as source files, binary files, object libraries, WYSIWYG documents, spreadsheets and anything which can be stored in the UNIX file system.
- Can be mounted on some or all workstations
- Several VOBs may exist on a machine or on different machines on a network.
- When mounted as a file system of type MFS, a VOB can be accessed with standard UNIX and ClearCase™ tools.
- The ClearCase™ file system is transparent.

A VOB is comprised of:

- Storage area for versioned files, derived objects and cleartext files.
- Database (live, shadow and log file).

A view is defined by the following characteristics:

- A working context for an individual developer or closely coordinated group.
- Can be used to access any VOB or multiple VOBs.
- Selects versions of VOB directories and files to display.
- Allows developer to work without interfering with other developers.
- Not a set of files but a way of seeing shared elements.
- Each user may have multiple views for new development, bug fixing or porting activities.

A view is comprised of:

- View storage area (typically in a local machine) - private storage for checked-out files, derived objects and private files.
- Configuration Specification - set of rules which determine the version of a file the view will see.
- View-tag - Name given to the view (ex. angies_view), view-tags are registered in /urs/adm/atria/view_tags.

Objects stored in a view:

- Checked-out versions of file elements.
- View private files.
- Unshared derived objects.

Creating a View in ClearCase™

The following procedure not only will create a view, but will also allow creation of a subdirectory where new science software files may be stored.

Assumptions:

- ClearCase™ is available.
- A Versioned Object Base (VOB) has been created.

Selecting a view listed in the View Tag Browser screen brings up the File Browser, or main screen, shown in Figure 3.

- Displays the directory name of the current VOB, just below the toolbar. A VOB is defined by several characteristics:
    - A mountable file system to store version-controlled data (e.g., source files, binary files, object libraries, spreadsheets).
    - Can be mounted on some or all workstations.
    - Several VOBs may exist on a machine or network.
    - When mounted as a file system of type MFS, a VOB can be accessed with standard UNIX and ClearCase™ tools.
    - The ClearCase™ file system is transparent.
- A VOB consists of a storage area (for versioned files, derived objects, and cleartext files) and a database (live, shadow, and log file).
- Displays the content of the directory in the space below the directory's name.
- Many ClearCase™ functions can be initiated from the File Browser screen.

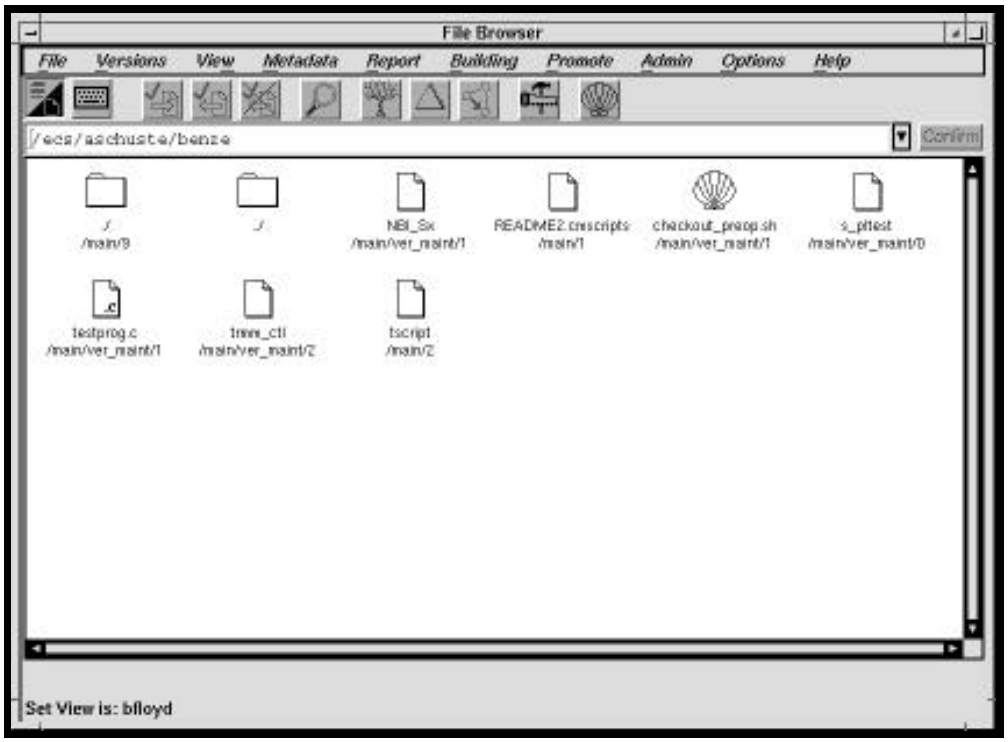

Figure 3. ClearCase™ File Browser Screen (Main Screen)

Creating a View in ClearCase™

---

1 The DAAC System Administrator should log into the aitn1sun or aitn2sun workstation by typing: **username** and then pressing the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** and then press the **Enter** key.

3 Invoke ClearCase™ by typing **xclearcase &** on the UNIX command line and then pressing the **Enter** key.

- The ClearCase™ **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase™ **View Tag Browser** screen is displayed listing available views.

4 To create a view for checking in the software change package, select a known View and press the **Enter** key.

- The File Browser window is displayed.

5 Select **File→Execute→Single Command.**

- The String Browser window is displayed.

- The prompt Enter shell command to run is displayed.

6 Invoke the make view command by typing **mkview [filename]** on the UNIX command line and pressing the **Enter** key.

- The **tempdisp** window appears.
- The **View [filename] Created Successfully** and the **Cache Updated for View [filename]** prompts are displayed.

7 Close the **tempdisp** window by clicking on the window and pressing the Enter key.

- The **tempdisp** window closes.

8 Select **View →List** from the menu.

- The **View Tag Browser** is displayed.

9 Find the new view by scrolling through the list until the new view is observed.

Setting a View in ClearCase™

In order to make files and directories that are in a ClearCase VOB (Versioned Object Base) visible and accessible, a ClearCase view must be set. Only one view can be set (active) at a time.

Setting a View in ClearCase™

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** and then pressing the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** and then press the **Enter** key.

3 Invoke ClearCase™ by typing **xclearcase &** on the UNIX command line and then pressing the **Enter** key.

- The ClearCase™ **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase™ **View Tag Browser** screen is displayed listing available views.

4 To set a view, select a known View and press the **Enter** key.

- The File Browser window is displayed.

5 Select **File→Execute→Single Command.**

- The String Browser window is displayed.
- The prompt Enter shell command to run is displayed.

6 Invoke the set view command by typing **setview ViewName** on the UNIX command line and pressing the **Enter** key.

- **ViewName** is the name of the view to set.

Creating a New Directory

In cases where a new directory needs to be created and placed in ClearCase™, the System Administrator will activate ClearCase™ and create a new directory. This type of procedure is necessary only if a new directory is required.

The following is a list of tools, and or assumptions:

- A VOB has been created at the UNIX directory.
- The System Administrator is the VOB administrator.
- A view has been created.

Entering a New Directory into ClearCase™

---

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** and then pressing the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** and then press the **Enter** key.

3 Invoke ClearCase™ by typing **xclearcase &** on the UNIX command line and then pressing the **Enter** key.

- The ClearCase™ **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase™ **View Tag Browser** screen is displayed listing available views.

4 Select **File→Execute→Single Command.**

- The String Browser window is displayed.
- The prompt **Enter shell command to run** is displayed.

5 Invoke the make directory element by typing **mkdir [filename]** on the UNIX command line and pressing the **Enter** key.

6 Invoke the make element command by typing **mkelem [directory name]** on the UNIX command line and pressing the **Enter** key.

7 Type into the directory input box of the **File Browser** the name of the directory in the VOB to be checked out and then press the **Enter** key, then follow the menu path **Version→Checkout→Reserved: no comment**.

- In order to add new files to ClearCase™, the directory in which the files are to be added must be checked out first.
- ClearCase™ forces the checkout onto a maintenance branch to isolate the maintenance activity.

- If someone else has already checked out the directory, permission to check out the directory is denied.
- A separate shell window is displayed.

8 Cancel the checkout of the element if it is decided that no changes are to be made by typing into the directory input box of the **File Browser** the name of the directory to be checked in and then press the **Enter** key, then follow the menu path **Version→Uncheckout→Unreserved: no comment**,

9 On the **File Browser** screen, follow the menu path **File→Exit**.

- The ClearCase™ Graphical User Interface session is closed.

Importing files into ClearCase™

Once the System Administrator has created a directory to place the science software files, ClearCase™ can be used to place a single file or multiple files in a UNIX directory structure under CM.

The following is a list of tools, and or assumptions:

- The DAAC System Administrator will be required in order to complete this procedure.
- A VOB and subdirectory are created to hold these files.
- No object files or executables exist in the source code directory.
- The PGE was received with a directory structure that contains various types of files.
- These files will be entered into ClearCase™ and will maintain the same directory structure as the delivery structure.

The first procedure to review will be importing a single file. The following procedure review covers importing multiple files in a UNIX directory structure.

Importing a Single File into ClearCase™

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** and then pressing the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** and then press the **Enter** key.

3 At a UNIX prompt, type **cleartool setview *ViewName***, press **Enter**

- The ***ViewName*** is the name of the ClearCase view (see Section 6.1 to create a view). For example, **cleartool setview jdoe,** press **Enter**.

4 At the UNIX prompt, type **cd *pathname***, press **Enter**.

- The ***pathname*** is the full path name of the subdirectory in the VOB into which the file is to be checked in. For example, to check a file into the VOB directory /VOB1/pge2/scripts/, type **cd /VOB1/pge2/scripts/**, press **Enter**. If the desired directory cannot be seen, it could mean that the view has not been set or the properties of the view do not allow the directory to be seen; check with the VA.

5 At a UNIX prompt, type **cp *pathname/filename* .**, press **Enter** (note the space and then "dot" at the end of the command)**.**

- The ***pathname*** is the full path name to the directory where the file to be checked in exists and ***filename*** is the file name of the file to be checked in. This command copies a file over into the VOB area in preparation for checking it in. For example, to copy over a file named MISR_calib.c in directory /pge/pge34/ to be checked in, type **cp pge/pge34/MISR_calib.c .**, press **Enter** (again, note the space and then "dot" at the end of the command).

6 At the UNIX prompt, type **cleartool checkout -nc .**, press **Enter** (note the space and then "dot" at the end of the command).

- This command checks out the current directory (represented by the "dot") from ClearCase. Adding a new file (or element) to a directory represents a modification of the directory. Hence, the directory must be checked out before a file can be checked in.

7 At a UNIX prompt, type **cleartool mkelem -nc *filename***, press **Enter.**

- The ***filename*** is the name of the file that was copied over in step 4 and is the file that will be checked into ClearCase. This command creates a ClearCase element from the file in preparation for checking it in. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the make element step.

8 At the UNIX prompt, type **cleartool checkin -nc *filename***, press **Enter**.

- The ***filename*** is the name of the file to be checked into ClearCase. This command performs the check in of the file. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check in step.

9 At the UNIX prompt, type **cleartool checkin -nc .**, press **Enter** (note the space and then "dot" at the end of the command).

- This command checks in the current directory (represented by the "dot") into ClearCase. The adding of an element (here, a file) represents a modification to the directory and hence, the new version of the directory must be checked back in. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check in step.

Importing Multiple Files into ClearCase™

1 The DAAC CM Administrator should log into the aitn1sun or aitn2sun workstation by typing: **username** and then pressing the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** and then press the **Enter** key.

3 Mount the software change package delivery medium (e.g., CD ROM) and put the package on the system by typing **tar xvf /dev/cd0** on the UNIX command line.

- The Delivered Algorithm Package directory structure is copied onto the system.

4 To go to the parent directory of the UNIX directory structure to be brought into ClearCase™ (i.e., the software change package), type **cd parentdir** (where **parentdir** is the path of the directory that contains the directory structure to be brought into ClearCase™) and then press the **Enter** key.

- This directory must not be in the VOB.

5 Create a conversion script by typing **clearcvt_unix -r dirname**, where **dirname** is the name of the directory containing the directory structure to be brought into ClearCase™ (i.e., the software change package).

- The command "clearcvt_unix" is a custom command that creates a script to check all the files in **dirname** into the VOB.
- The addition of "-r" to the command ensures that any subdirectories below **dirname** will be recursively included in the script created.

6 At this time the DAAC CM Administrator logs out from this workstation. The DAAC System Administrator (SA) completes the procedure.

- The remaining steps are accomplished by the DAAC SA.

7 The DAAC SA logs into the aitn1sun or aitn2sun workstation by typing **username** and then pressing the **Enter** key.

- Cursor moves to the **Password** field.

8 Type the **password** and then press the **Enter** key.

9 Invoke Clear Case™ by typing **xclearcase &** on the UNIX command line and then pressing the **Enter** key.

- The ClearCase™ **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase™ **View Tag Browser** screen is displayed listing available views.

10 To create a view for checking in the software change package, select a known View and press the **Enter** key. If you are using an existing view, select the desired existing view and proceed to step 14.

- The File Browser window is displayed.

11 Select **File→Execute→Single Command.**

- The String Browser window is displayed.
- The prompt Enter shell command to run is displayed.

12 Invoke the make view command by typing **mkview [filename]** on the UNIX command line and pressing the **Enter** key.

- The **tempdisp** window appears.
- The **View [filename] Created Successfully** and the **Cache Updated for View [filename]** prompts are displayed.

13 Close the **tempdisp** window by clicking on the window and pressing the Enter key.

- The **tempdisp** window closes.

14 VOB where the software change package is to be imported and then press the **Enter** key. Then, to create a subdirectory for the software change package in that VOB, which is a modification to the parent directory (for the VOB) that requires the parent directory to be checked out first, then follow the menu path **Version→Checkout→Reserved: no comment**.

- In order to add new files to ClearCase™, the directory in which the files are to be added must be checked out first.
- ClearCase™ forces the checkout onto a maintenance branch to isolate the maintenance activity.
- If someone else has already checked out the directory, permission to check out the directory is denied.
- A separate shell window is displayed.

15 To prepare to run the conversion script that will check in the files from the software change package, start a shell process in a separate window by clicking on the shell icon button of the **File Browser** toolbar.

- A separate shell window is displayed.

16 To run the script, type **cvt_script** and then press the **Enter** key.

- The SA is the only person who can run the **cvt_script** because it modifies the VOB. All VOB control is kept by the SA for security purposes.

17 To check in the new directory, type into the directory input box of the **File Browser** screen: **path** [where **path** is the full path identification for the new directory (**directoryname**)], and then press the **Enter** key. Then follow menu path **Versions→Checkin**.

18 To check in the parent directory (for the VOB), type into the directory input box of the **File Browser** screen: **VOBpath** (where **VOBpath** is the full path identification for the parent directory, and then press the **Enter** key. Then follow the menu path **Versions→Checkin**.

19 On the **File Browser** screen, follow menu path **File→Exit**.

- The ClearCase™ Graphical User Interface session is closed.

Checking Out a File From ClearCase™

If a configured file requires modification, then the file needs to be checked out of the configured directory and placed in a user directory. This will allow the file(s) to be modified.

The following is a list of tools, and or assumptions:

- The file or directory must be an element created in ClearCase™.
- The view should be configured to ensure the correct version of the file or directory is seen.

Checking Out an Element/File

---

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** and then pressing the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** and then press the **Enter** key.

3 Invoke ClearCase™ by typing **xclearcase &** on the UNIX command line and then pressing the **Enter** key.

- The ClearCase™ **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase™ **View Tag Browser** screen is displayed listing available views.

5 To check out the directory where the controlled files were place, type into the directory input box of the **File Browser** screen: **path** [where **path** is the full path identification for the directory (**directoryname**)], and then press the **Enter** key. Then click on the directory to be checked out and select **Versions→Checkout** from the menu.

6 To check out a controlled file, click on the file to be checked out and select **Versions→Checkout** from the menu.

7 Select **File→Execute→Single Command.**

- The String Browser window is displayed.
- The prompt **Enter shell command to run** is displayed.

8 To determine editing privileges, type **ls -l** then press the **Enter** key.

- A prompt displaying read/write/execute privileges will be displayed. There will be three groupings:

  User Group Others

  _ _ _ _ _ _ _ _ _ **r=read, w=write, x=execute**

9 If you have editing/execute privileges, you can revise the contents of the file with any text editor.

10 To checkin a controlled file, select **Versions→Checkin** from the menu.

- The file/directory will be checked in to ClearCase™ and the version will be updated.

Checking a Modified Element into ClearCase™

This procedure explains how to check in a modified element to ClearCase. An element refers to a directory or file in ClearCase, that is, under configuration management. Modifications made to a file or directory cannot be saved in ClearCase unless the file or directory had been checked out first. See Section 6.6 on how to check out an element.

The assumptions are that a VOB exits and is mounted at a known UNIX directory. A ClearCase view exists for the SSI&T operator.

Importing a Single File into ClearCase™

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** and then pressing the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** and then press the **Enter** key.

3 At a UNIX prompt, type **cleartool setview *ViewName***, press **Enter**

- The ***ViewName*** is the name of the ClearCase view (see Section 6.1 to create a view). For example, **cleartool setview jdoe,** press **Enter**.

2 At the UNIX prompt, type **cleartool checkin -nc *element***, press **Enter**.

- The ***element*** is the name of the file or directory (full path name allowed) that is to be checked out (and later modified). The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check out step.

3 This step is optional; it is performed when ClearCase does not accept a check in because the element was not modified. In this case, the check out must be canceled. At a UNIX prompt, type **cleartool uncheckout -nc *element***, press **Enter.**

- The ***element*** is the name of the file or directory (full path name allowed) checked out. This command cancels the check out of an element. The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the uncheckout step.

This page intentionally left blank.

Preparation of Earth Science Data Types

Every science data product generated and archived by the ECS must be described to the system by metadata which are put into an inventory and then used to retrieve and distribute the data to users of the system. The ECS Earth Science Data Model organizes the metadata into groups of related attributes and services to be performed on the data products. Granules of the same type of science data are grouped into collections. Every collection is described by an Earth Science Data Type (ESDT) and is made known to the system by an ESDT descriptor file and associated software code which is built into the Data Server's dynamic link library (DLL) to perform the services. The ESDT descriptor is composed of sections containing the following information:

- Collection level metadata attributes with values contained in the descriptor.
- Granule level metadata attributes whose values are supplied primarily by the Product Generation Executives (PGEs) during runtime.
- Valid values and ranges for the attributes.
- List of services for the data and events which trigger responses throughout the system.

The ESDTs for all data collections to be input to or output from the PGEs must be built and registered into the ECS before any of the PGEs can be run under the automated processing system. Since the original Release A Data Server is no longer a part of the Pre-Release B Testbed, the DLL code is no longer required.

During the past year, the ECS has collected information from the Instrument Teams on the ESDTs needed for their science software in the Pre-Release B Testbed. This information has been baselined and a set of ESDT descriptor files have been built and tested according to this baseline. The baselined ESDT descriptor files reside in the Testbed under ECS configuration management in a VOB called the ECS Configuration Area. Since science software may have evolved during the time between the ESDT baseline and delivery to the DAAC, some changes to the baselined ESDT descriptor files should be expected. In addition, some new ESDTs may be required which were not included in the ESDT baseline.

A set of SSI&T ESDT Tools have been developed to automate the process of making changes to existing ESDTs. The procedures and tools that follow describe how to assess whether changes have been made relative to the baselined ESDTs and if so, how to make new versions of these ESDTs. Another tool allows the generation of new ESDTs from scratch.

Comparing Granule Level Metadata

A PGE accesses granule level metadata attributes and values via a Metadata Configuration File (MCF). There is typically one MCF for each output data set. The ESDTs which have been built and registered in the ECS contain a section for granule level metadata attributes and values for

each data set. In terms of content, the MCFs and the granule level metadata section of the corresponding ESDT descriptor files have to be in sync. As expressed in the introductory paragraphs (above), the science software in the PGEs may have changed to a point where these two locations of granule level metadata are no longer in sync.

Few changes are expected in the Inventory section of the MCF. Changes are more likely to be expected in the Archive section of the MCF and in the Product Specific Metadata. If there are *any* changes, a new version of the baselined ESDT descriptor file must be generated.

Section 5.1.1 describes how to determine if the delivered MCF and the registered ESDT descriptor file are in sync with one another. Section 5.1.2 describes how to generate a new version of an existing ESDT descriptor file if the delivered MCF is found not to be in sync with the ESDT descriptor file.

Comparing Granule Level Metadata in ESDT to Delivered MCF

The MCFToDescChecker tool compares a delivered Source MCF with the granule level metadata section of an ESDT descriptor file. The program requires as input the delivered Source MCF and a copy of the existing ESDT descriptor from the ECS Configuration Area under ClearCase. Execution of this program will indicate whether the granule level metadata content is consistent between the MCF and the ESDT descriptor file or it will produce diagnostics indicating where differences have been detected.

The procedures assume that the Source MCF is available and accessible and that there exists in the ECS Configuration Area an ESDT corresponding to this MCF. It is further assumed that the VA has granted ClearCase privileges for the user.

To compare the granule level metadata in the ESDT to the delivered MCF, execute the procedure steps that follow:

---

1 At a UNIX prompt on the AIT Sun, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of the view that allows access to the ECS Configuration VOB to be set.
- For example, type **cleartool setview ecsconfig**, press **Enter**.

2 At a UNIX prompt on the AIT Sun, type **MCFToDescChecker *pathname*/*ESDTfilename MCFfilename,*** press **Enter**.

- The ***pathname*** is the full path name to the location of the ESDT descriptor file (in the VOB).
- The ***ESDTfilename*** is the file name of the ESDT descriptor file. By convention, the file name is the ESDT ShortName.

- The ***MCFfilename*** is the file name of the delivered Metadata Configuration File (MCF) with which to compare the ESDT descriptor file.
- For example, type **MCFToDescChecker /ecs/formal/PDPS/Rel_A/DPS/testbed/ESDT/CER/CER01aT CER01aT.MCF**
- After the program has run, a `Identical` or `Not Identical` will be displayed. A `Identical` indicates that the MCF and ESDT descriptor file granule level metadata were identical in content; a `Not Identical` indicates a discrepancy.

Generating a New Version of ESDT from Delivered MCF

If the delivered Source MCF is found to be incompatible with the baselined ESDT descriptor file, the UpdateDesc tool can be used to update the original ESDT descriptor file to a new version. This new version will have the same ShortName, but a new VersionID. The UpdateDesc tool requires as input the ESDT descriptor file from the ECS Configuration Area and the delivered Source MCF. The new ESDT version will be constructed with a granule level metadata section that is compatible with the delivered MCF. To verify that the new ESDT descriptor file has been correctly updated, the MCFToDescChecker tool (see Section 5.1.1) can be run again using the new ESDT descriptor file.

The procedures assume that the Source MCF is available and accessible and that there exists in the ECS Configuration Area an ESDT corresponding to this MCF. It is further assumed that the VA has granted ClearCase privileges for the user.

To generate a new version of an ESDT from the delivered MCF, execute the procedure steps that follow:

---

1 At a UNIX prompt on the AIT Sun, type **cleartool setview *viewname***, press **Enter**.

- The ***ViewName*** is the name of the view that allows access to the ECS Configuration VOB to be set.
- For example, type **cleartool setview ecsconfig**, press **Enter**.

2 At a UNIX prompt on the AIT Sun, type **cleartool checkout -nc *pathname*/*ESDTfilename***, press **Enter**.

- The ***pathname*** is the full path name to the location of the ESDT descriptor files.
- The ***ESDTfilename*** is the file name of the ESDT descriptor file that will be modified.
- The **-nc** stands for "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check out step.
- For example, type **cleartool checkout -nc /ecs/formal/PDPS/ESDT/CER/CER01aT**

3 At a UNIX prompt on the AIT Sun, type **UpdateDesc *ESDTfilename MCFfilename ESDTfilename*.new,** press **Enter**.

- The ***ESDTfilename*** is the file name of the ESDT descriptor file to be modified.
- The ***MCFfilename*** is the file name of the delivered Metadata Configuration File (MCF).
- The ***ESDTfilename*.new** is the file name to be given to the output ESDT descriptor file. The file name of the ESDT descriptor file, by convention, is the ESDT ShortName. The file name extension .new indicates that it is a new version.
- For example, type **UpdateDesc CER01aT CER01aT.MCF CER01aT.new**, press **Enter.**

4 At a UNIX prompt on the AIT Sun, type **MCFToDescChecker *ESDTfilename*.new *MCFfilename***, press **Enter**.

- The ***ESDTfilename*.new** is the file name of the modified ESDT descriptor file created in step 5.
- The ***MCFfilename*** is the file name of the delivered Metadata Configuration File (MCF).
- This step verifies that the new ESDT descriptor file has identical granule level metadata content identical to the delivered MCF and that the Update Descriptor File Tool was successful.
- For example, type **MCFToDescChecker CER01aT.new CER01aT.MCF**, press **Enter.**

5 If the **MCFToDescChecker** returned a `Identical`, at a UNIX prompt on the AIT Sun, type **mv *ESDTfilename*.new *ESDTfilename***, press **Enter**.

- The ***ESDTfilename*.new** is the file name of the modified ESDT descriptor file created in step 4.
- The ***ESDTfilename*** is the file name of the original ESDT descriptor file and the file name to be given to the modified version.
- This command renames the modified version of the ESDT descriptor file to the original file name.
- For example, type **mv CER01aT.new CER01aT**, press **Enter**.
- Depending upon the user environment, the shell may ask for confirmation since the original version of the descriptor file will be overwritten.

6 At a UNIX prompt, type **cleartool checkin -c "*comment*" *pathname*/*ESDTfilename,*** press **Enter.**

- The ***pathname*** is the full path name to the location of the ESDT descriptor files in the ClearCase VOB. The directory must be checked out before the file to modify.
- The ***ESDTfilename*** is the name of the now modified ESDT descriptor file.
- The **-c "*comment*"** indicates a comment to be associated with the check in. The comment must be enclosed within double quotes and immediately follow the **-c** flag.
- This command checks in the modified ESDT descriptor file to ClearCase.
- For example type, **cleartool checkin -c "Modified according to delivered MCF at Pre-Release B Testbed" /ecs/formal/PDPS/ESDT/CER/CER01aT,** press **Enter**. Alternatively, use the **-nc** flag to enter in no comment.

Creating a New ESDT

In some cases, PGEs will require ESDTs that are not part of the ESDT baseline and are therefore not available at the DAAC. In these cases, new ESDTs will have to be built from scratch. The tools provided for generating new ESDTs assume limited collection level metadata (six attributes). These tools can generate ESDTs for dynamic data granules (those that are associated with a temporal locality) or static data granules (typically, permanent files like lookup tables or coefficient files).

Generation of a new ESDT requires several steps. First, a comma-delimited text file containing the six collection level attribute values is needed. Such a file is most easily generated using a spreadsheet program like MS Excel. This procedure is described in Section 5.2.1.

Second, the MCF corresponding to the new ESDT is required. For dynamic granules, at least, this should be part of the delivery to the DAAC. If the data granule is static, however, then it is unlikely that a corresponding MCF has been delivered. In this case, a default template MCF will be assumed. These procedures are described in Section 5.2.2.

If ESDTs require Product Specific Attributes (PSAs), these will have to be added to the generated ESDT descriptor file manually after it is been constructed.

Every file that is to be Inserted to or Acquired from the IMF Data Server must have some ESDT defined for it.

Generating a Comma-Delimited Text File from a Spreadsheet

In order to generate a new ESDT descriptor file, a comma-delimited text file must be created. Such a file can be created from a MS Excel spreadsheet.

The procedures assume that the SSIT Manager is running. This is required to bring up the MS Windows emulator and then MS Excel.

To generate a comma-delimited text file from a spreadsheet, execute the procedure steps that follow:

---

1 From the SSIT Manager, click on the **Tools** menu, then choose **Office Automation** and then **MSWindows**.

- The MS Windows emulator GUI will be displayed.

2 Within the MS Windows GUI, double click on the MS Excel icon.

- The MS Excel program will start and a blank spreadsheet will be opened.

3 In the MS Excel GUI, enter in the following column headings into cells A1, B1, C1, D1, E1, F1, and G1: `ShortName`, `LongName`, `VersionID`, `CollectionDescription`, `ProcessingCenter`, `ArchiveCenter`, `Permanent`

- Enter the column headings as shown, one per cell in the default font.
- The tab key may be used to move the cursor from one cell to the next cell to the right.

4 Beginning in row B, populate the spreadsheet with values corresponding to the column headings entered above. These column headings refer to collection level metadata (a minimum set) that are necessary to later create an ESDT descriptor file. Each row entered will correspond to one ESDT to be created.

- `ShortName` must be a string less than or equal to eight characters.
- `LongName` must be a string less than or equal to 80 characters.
- `VersionID` must be a string less than or equal to 20 characters.
- `CollectionDescription` must be a string less than or equal to 255 characters.
- `ProcessingCenter` must be set to one of: `EDC`, `GSFC`, `LaRC`, or `NSIDC`.
- `ArchiveCenter` must be set to one of: `EDC`, `GSFC`, `LaRC`, or `NSIDC`

- `Permanent` must be set to one of: `Yes` or `No`. A `Yes` indicates that a MCF is **not** available for making an ESDT for this entry (*e.g.* because the file is static); a `No` indicates that a MCF **is** available for making an ESDT for this entry (the MCF will be used later).

- Information for populating these fields should be provided by the Instrument Teams.

- Do **not** insert comments or superfluous information between the column headings and the data or within the data area.

5 Once the spreadsheet has been populated with collection level metadata information for each ESDT to generate, click on the **File** menu, then choose **Save As…**.

- This will bring up a dialog box labeled **Save As**.

6 In the **Save As** dialog box, enter a file name in the field next to **File name:** under which to save the comma-delimited text file. Then in the **Save as type:** menu, select **CSV (comma delimited)(*.csv)** item. Finally, click on the **Save** button.

- Optionally, the field next to **Save in** may be used to specify a directory in which to save the file.

- The comma-delimited text file version of the spreadsheet created will be saved to local disk.

7 In the MS Excel GUI, click on the **File** menu, then choose **Exit**.

- This will close MS Excel.

- Optionally, quit MS Windows by clicking on the **File** menu in the Program Manager GUI, then choose **Exit**. A confirmation dialog box will be displayed; choosing **OK** will end the session with MS Windows.

Generating a New ESDT Descriptor File

The DescGen tool provides the capability to generate new ESDT descriptor files with limited collection level metadata only. This program is written in the C language and is designed to be run from the command line to produce one or more new ESDT descriptor files depending on the number of rows of *ShortName* for the collection and other attributes supplied in the metadata text file (see Section 5.2.1). The new ESDT descriptor files are then checked into ClearCase in the ECS Configuration VOB. DescGen can be packaged into a script with the ClearCase commands for a completely automated procedure for creating and configuring multiple new descriptor files. DescGen requires several input files containing metadata and several templates for the ESDT descriptor syntax and format required to produce the output ESDT descriptor files.

The following list contains the input files required for DescGen.

- The text file containing limited collection level metadata created in Section 5.2.1. DescGen reads the attribute values from this comma delimited text file and inserts them into the collection metadata section of the ESDT descriptor template. The value in the last column indicates whether the ESDT descriptor is to be generated for a static or permanent file requiring no MCF for the program input or a dynamic data granule for which an MCF must be provided as program input.

- The MCFs which are delivered with the PGE for new output products. The MCFs contain the granule level attributes which DescGen will insert into the granule level metadata section of the ESDT descriptor template. The program assumes that there is a MCF with the file name *ShortName.*MCF for each ESDT to be generated.

The following list contains the input templates required for DescGen.

- TemplateColl.odl is an ESDT descriptor template file defining limited collection level metadata attributes for dynamic data granule type of files for which an MCF with the granule level attributes will be provided as input to the program.

- TemplateCollYes.odl is an ESDT descriptor template file for limited collection level metadata attributes for static or permanent type of files for which no MCF will be provided as input to the program.

- TemplateService.odl is a template for the Service Group in the ESDT descriptor file. It will contain only generic services provided by ECS at the Pre-Release B Testbed.

The following paragraph describes the output file from DescGen.

- DescGen will generate one output ESDT descriptor file for each row in the collection level metadata text file. By convention at the Testbed, the ESDT descriptor file name is *ShortName*.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

The procedures assume that all of the values for the collection level attributes and the type of file (static or dynamic data granule) in the spreadsheet have been provided by the Instrument Teams for each new ESDT to be generated and that the spreadsheet has been saved as a comma delimited text file. The procedures further assume that both the comma delimited text file and all of the corresponding required MCFs are in the current Instrument SSI&T work directory. If product specific metadata are requested for any of the ESDTs and granule level product specific attributes are included in any of the MCFs, these attributes will have to be entered into the ESDTs after the new ESDTs have been created in step 1 by using the text editor in step 2.

To generate an ESDT descriptor file, execute the procedure steps that follow:

1 At a UNIX prompt on the AIT Sun, type **DescGen *TextFilename***, press **Enter.**

- The ***TextFilename*** is the file name of the collection level metadata text file in comma-delimited format created in Section 5.2.1.
- For example, type **DescGen CERESDT.csv**, press **Enter**.

2 At a UNIX prompt on the AIT Sun, type **vi *DescFilename***, press **Enter.**

- The ***DescFilename*** is the file name of the ESDT descriptor file created in step 1. The file name will be the ESDT's ShortName.
- For example, type **vi CER01aT**, press **Enter**.
- Any text editor may be used such as *emacs*. For example, **emacs CER01aT**, press **Enter**.
- In the editor, review the file then quit the editor.

3 At a UNIX prompt on the AIT Sun, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of the view that allows access to the ECS Configuration VOB to be set. For example, type **cleartool setview ecsconfig**, press **Enter**.

4 At a UNIX prompt on the AIT Sun, type **cd *ESDTpathname***, press **Enter**.

- The ***ESDTpathname*** is the full path name of the subdirectory in the ECS Configuration VOB containing the ESDTs for the relevant instrument.
- For example, type **cd /ecs/formal/PDPS/Rel_A/testbed/ESDT/CER**, press **Enter**.

5 At a UNIX prompt on the AIT Sun, type **cp *pathname*/*DescFilename* .**, press **Enter** (note the space and then "dot" at the end of the command).

- The ***pathname*** is the full path name to the directory containing the new ESDT descriptor file.
- The ***DescFilename*** is the file name of the new ESDT descriptor file created in step 1. The file name will be the ESDT's ShortName.
- For example, type **cp /SSIT/CERES/ESDT/CER01aT .**, press **Enter** (note the space and then "dot" at the end of the command).

6 At a UNIX prompt on the AIT Sun, type **cleartool checkout -nc .**, press **Enter** (note the space and then "dot" at the end of the command).

- This command checks out the current directory (represented by the "dot") from ClearCase. Adding a new file (or element) to a directory represents a modification of the directory. Hence, the directory must be checked out before a file can be checked in.
- The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check out step.

7 At a UNIX prompt on the AIT Sun, type **cleartool mkelem -nc *DescFilename***, press **Enter.**

- The ***DescFilename*** is the name of the new ESDT descriptor file that was copied over in step 5 and is the file that will be checked into ClearCase. This command creates a ClearCase element from the file in preparation for checking it in. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the make element step.
- For example, type **cleartool mkelem -nc CER01aT**, press **Enter**.

8 At the UNIX prompt on the AIT Sun, type **cleartool checkin -c "comment" *DescFilename***, press **Enter**.

- The ***DescFilename*** is the name of the file to be checked into ClearCase. This command performs the check in of the file.
- The **-c "*comment*"** indicates a comment to be associated with the check in. The comment must be enclosed within double quotes and immediately follow the **-c** flag.
- For example, type **cleartool checkin -c "New ESDT generated for the Pre-Release B Testbed" CER01aT**, press **Enter**.

9 At the UNIX prompt on the AIT Sun, type **cleartool checkin -nc .**, press **Enter** (note the space and then "dot" at the end of the command).

- This command checks in the current directory (represented by the "dot") into ClearCase. The adding of an element (here, a descriptor file) represents a modification to the directory and hence, the new version of the directory must be checked back in. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check in step.

Registering an ESDT

The tool test_esdt provides the capability to register a new ESDT in the IMF Data Server of the Pre-Release B Testbed. The tool is run from the command line by authorized ECS SSI&T or Systems Administrator personnel. The program creates a new directory for the ESDT in the ECS PDPS configured area. If the ESDT collection is generated for a dynamic science data granule, the PDPS will put into this directory the ESDT descriptor file (with the ESDT's ShortName used as the file name), the science data granule files, and the Target MCFs (*ShortName*.met files) which are used for Inserting the granules to the IMF Data Server. If the ESDT collection is generated for static or permanent data files, the PDPS will put into this directory the ESDT descriptor file (again, with the ESDT's ShortName used as the file name) and the static files. By convention, a container or bucket ESDT is generally created for each PGE to hold all static or permanent files needed by the PGE.

The following list contains the information required for input to the test_esdt tool.

- The archive name which will be supplied by ECS. The archive name, in the IMF, refers to the UNIX path name of the archive.
- The ShortName for the ESDT.
- The ESDT descriptor file name. In the Pre-Release B Testbed IMF, the descriptor file name must be the same as the ShortName.

The procedures assume that the ECS Configuration Area is properly setup, that the path name is known to SSI&T personnel, and that the SSI&T personnel have permissions and privileges to register an ESDT into the ECS.

To register an ESDT, execute the procedure steps that follow:

---

1 At the UNIX prompt on the AIT Sun, type **test_esdt**, press **Enter**.

2 At the program prompt **Archive Name>**, type ***pathname***, press **Enter**.

- The ***pathname*** is the full path name to the IMF archive in which to register the ESDT.
- For example, type **/ECS/Rel_A/PDPS/archive**, press **Enter**.

3 At the program prompt **ShortName>**, type ***ShortName***, press **Enter**.

- The ***ShortName*** is the ShortName of the ESDT.
- For example, type **CER01aT**, press **Enter**.

4 At the program prompt **Descriptor File>**, type ***DescFilename***, press **Enter**.

- The ***DescFilename*** is the file name of the ESDT descriptor file. This file name must be the same as the ShortName for the ESDT.
- For example, type **CER01aT**, press **Enter**.

5 Exit the program.

- The ESDT will then be registered.

Validating Successful ESDT Registration

After registering a new ESDT (see Section 5.3), a subdirectory should be created in the IMF Data Server archive which is located in an ECS PDPS configured area. The new ESDT subdirectory will be named with the ESDT's ShortName. The ESDT registration tool, test_esdt, will copy the corresponding ESDT descriptor file from the ESC configuration ClearCase VOB to the IMF Data Server archive in the corresponding ESDT subdirectory. When the registration process has completed, the SSI&T staff should validate its success. In the IMF Data Server, the criteria of success will be verifying that the new ESDT subdirectory has been created in the correct directory, that the new ESDT descriptor file has been copied into this subdirectory, and that the contents of the ESDT descriptor file are identical to the ESDT descriptor file in the ECS Configuration ClearCase VOB.

To validate successful ESDT registration, execute the procedure steps that follow:

1 At the program prompt on the AIT Sun, type **cd *ArchivePathname***, press **Enter**.

- The ***ArchivePathname*** is the full path name of the IMF Data Server archive.
- For example, type **cd /ECS/Rel_A/PDPS/archive**, press **Enter**.

2 At the program prompt on the AIT Sun, type **ls -al**, press **Enter**.

- A list of the current contents will be displayed. These will mostly be directories, one for each ESDT that currently exists. The directories will be named with the ESDT ShortName.

3 At the program prompt on the AIT Sun, type **cd *ShortName***, press **Enter**.

- The ***ShortName*** is the name of the subdirectory containing the new ESDT descriptor file.
- For example, type **cd CER01aT**, press **Enter**.

4 At the program prompt on the AIT Sun, type **ls -al**, press **Enter**.

- A list of the current contents of the ***ShortName*** directory will be displayed.

5 At a UNIX prompt on the AIT Sun, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of the view that allows access to the ECS Configuration VOB to be set. For example, type **cleartool setview ecsconfig**, press **Enter**.

6 At a UNIX prompt on the AIT Sun, type **diff *ShortName pathname/ShortName***, press **Enter**.

- The ***ShortName*** is the file name of the ESDT descriptor file that exists in both the IMF Data Server archive and in the ECS Configuration VOB.
- The ***pathname*** is the full path name to the location in the ECS Configuration VOB where the ESDT descriptor file resides under configuration management.
- For example, type **diff CER01aT /ecs/formal/PDPS/ESDT/CER/CER01aT**, press **Enter**.
- If no diagnostic messages are displayed, it means that the ESDT registration process was successful. If diagnostic message are displayed, this indicates some problem with the ESDT registration that should be investigated.

This page intentionally left blank.

Science Software Integration and Test (SSIT) Manager

SSIT Manager Overview

The SSIT Manager is the top level GUI presented to personnel involved in performing SSI&T. Its purpose is to bring together the tools needed for SSI&T. The SSIT Manager runs only on the Testbed AIT Suns.

It is recommended that the xterm window used to start the SSIT Manager not be used for any other purpose.

The SSIT Manager provides access to all COTS tools and custom applications that are part of the SSI&T environment to include the following operating functions:

- Prepare SSI&T checklist.
- Open xterm session.
- Code analysis used to debug problems.
- Office automation tools needed to view files, access the WWW, and perform documentation.
- Check for standards compliance.
- Product examination in HDF format.
- File comparison.
- Edit arbitrary text file.
- Initialize and update Planning/Production (PDPS) databases.
- Data server access.

The following assumptions are made with regard to the use of the SSIT Manager application.

- The operator is located at a workstation or server to which the SSIT Manager has been configured.
- The operator has proper authorization to access the PDPS/SSIT database and the Data Server.
- The operator is logged into DCE.

- To access files in ClearCase™, the operator has a ClearCase™ view already set.
- The operator's environment has been configured as documented in the file **$ECS_HOME/docs/README_SSIT_<mmddyy>** - use the file with the latest date.

SSIT Manager GUI

This GUI (Figure 1) is the starting point for SSI&T activities. It provides access to a collection of tools that will be useful for this purpose.

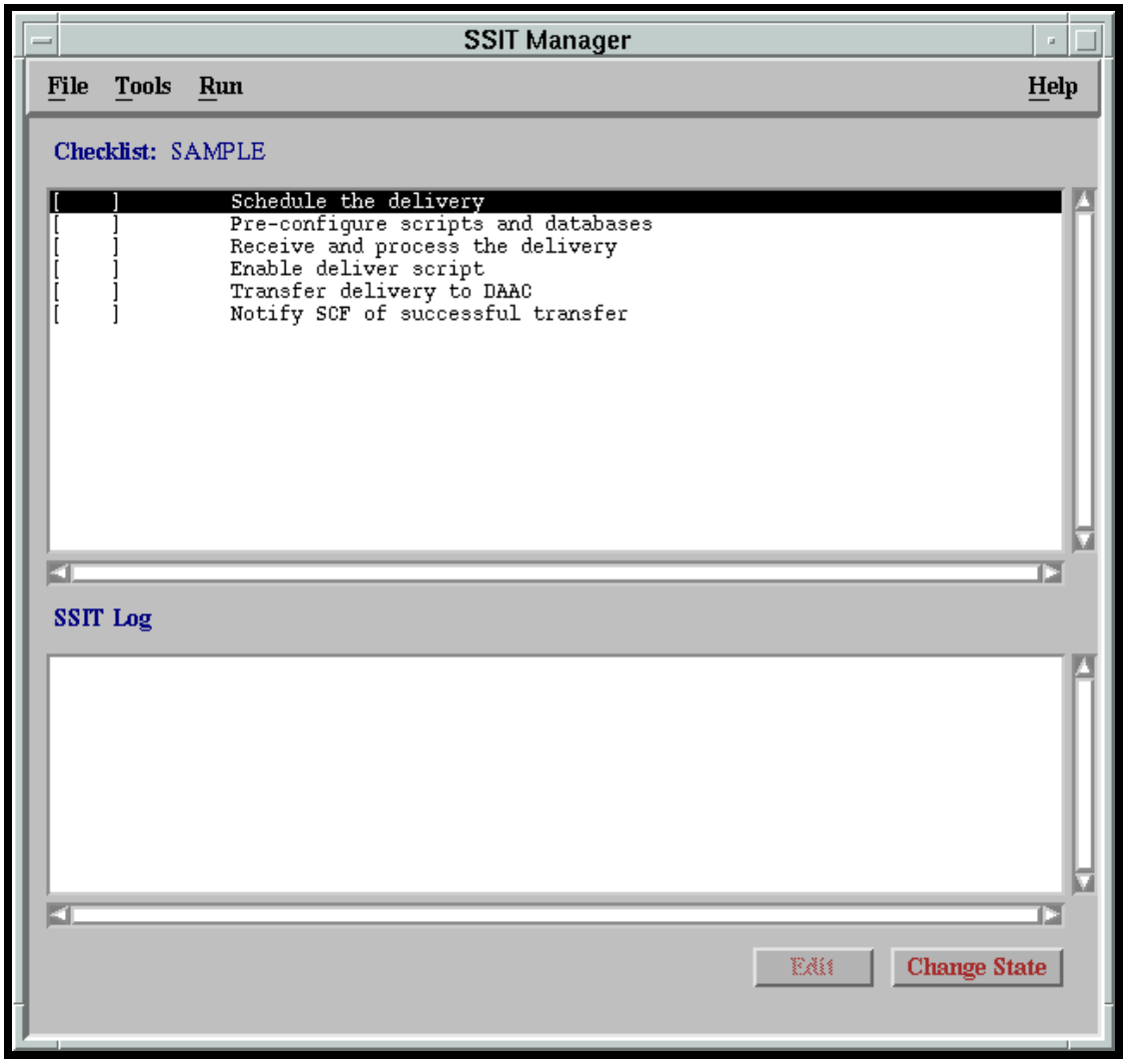

Figure 1. SSIT Manager Window

General Set Up of the SSIT Manager

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

- If you access the workstation through a remote login (rlogin), you must enter **xhost +** and then press the **Enter** key.
- Once you have entered xhost +, but prior to the remote login, enter **setenv DISPLAY** <local workstation IP address>**:0.0** where the local workstation IP address represents the IP address you where you are located**.**
- You may need to setup the terminal so that the remote host is displayed on your screen. (Sun machine) This is done by clicking on the **Application Manager** icon (the file drawer located at the bottom of the screen), followed by the **Desktop Tools** icon**,** followed by the **Terminal Consol** icon, then typing **xhost + aitn1sun** or **aitn2sun.gsfc.ecs.nasa.gov** in the display window.

3 At a UNIX prompt on the AIT Sun, type **cp *SSITpcfPathname*/*filename* $HOME/*mySSITpcf***, press **Enter**.

- The ***SSITpcfPathname*** is the full path name of the location of the SSIT Manager's default internal Process Control File (PCF).
- The ***filename*** is the file name of the PCF.
- The complete path and file name is usually EcsCustomSw/DpAtMgrInternal.pcf*machine_name*, where *machine_name* is the name of the machine on which the SSIT software was installed
- The ***mySSITpcf*** is the file name to the private copy of the PCF that the SSI&T operator will use when running the SSIT Manager. The **$HOME** is the environment variable for the user's home directory. For example, **cp /RelA/data/PCF.v5.ssit.dprs1gsfc $HOME/myPCF**, press **Enter**.

4 At a UNIX prompt on the AIT Sun, type **vi $HOME/.cshrc**, press **Enter.**

- This command invokes the *vi* editor and reads in the .cshrc file from the user's home directory.
- Any text editor may be used such as *emacs*. For example, **emacs $HOME/.cshrc**, press **Enter**.

5 In the file, add the following line if not already there: **setenv DPATMGR_HOME *SSITmanagerPathname***.

- The ***SSITmanagerPathname*** is the full path name to the home directory of the SSIT Manager on the AIT Sun. This information should be provided by the SA.

6 In the file, add the following line if not already there: **setenv PGS_PC_INFO_FILE $HOME/*mySSITpcf***

- The ***mySSITpcf*** is the full path name to the private copy of the PCF to be used with the SSIT Manager when you run it (from step 1).

7 In the file, add the following line if not already there: **setenv PGSHOME *ToolkitPathname***

- The ***ToolkitPathname*** is the full path name to location of the SDP Toolkit home directory on the AIT Sun. This information should be provided by the SA.

8 In the file, add the following line if not already there: **setenv PGSMSG $PGSHOME/message**

- The line must be after the line entered in step 5.

9 In the file, add the following line if not already there: **source $DPATMGR_HOME/bin/sun5/DpAtEnv.csh**

- This command causes the file DpAtEnv.csh to be run (sourced) and other environment variables to be set.

10 This step is optional. In the file, add the following line if not already there: **setenv DISPLAY *machinename*:0.0**

- The ***machinename*** is the name of the machine on which the SSIT Manager is to be displayed and operated. For example, if the machine name is spr1sgigsfc, then enter **setenv DISPLAY spr1sgigsfc**.
- If access to the SSIT Manager will not be from the same machine (all or most of the time), this command should not be added to the .cshrc. Instead, the operator will have to run the command from the command line at the UNIX prompt each time *before* the SSIT Manager is started.

11 Save the changes made to the .cshrc file and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Enter**.
- For other editors, refer to that editor's documentation.

12 At the UNIX prompt on the AIT Sun, type **source $HOME/.cshrc**, press **Enter**.

- This command causes the .cshrc file to get run and the new commands in it to be executed.
- Optionally, the user may logout and then log back in. The result will be the same as above.

Checklist for SSIT Manager

The SSIT Manager offers the capability of maintaining user-defined checklist of SSI&T activities. The checklist is presented in the main window of the SSIT Manager. This procedure explains how to set up this checklist.

Preparing and Updating a Checklist

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun, type **cp $DPATMGR_HOME/data/checklist.sample $HOME/*mychecklist***, press **Enter**.

- The ***mychecklist*** is the file name for a private copy of the checklist file. It will be this copy that will be edited in the steps which follow.
- **$DPATMGR_HOME** is an environment variable pointing to the home directory of the SSIT Manager.
- **$HOME** is the user's home directory. Although the checklist file does not have to be in the user's home directory, the following steps assume that it is.

4 At a UNIX prompt on the AIT Sun, type **vi ~/*mychecklist***, press **Enter.**

- The ***mychecklist*** is the file name of the checklist created in step 1. This command invokes the *vi* editor and reads in the checklist file from the user's home directory. Alternatively, any text editor may be used such as *emacs*. For example, **emacs ~/*mychecklist***, press **Enter**.

5 In the file, search for the lines of the form **DATABASE=*ssitUserPathname*/*filename*** and **CHECKLIST=*title***. Edit these lines.

- The ***ssitUserPathname*** is the full path name to where the database files (used in conjunction with the checklist) will be placed. Typically, this is the user's home directory.
- The ***filename*** is the base name for the database files. Two database files will be created in the directory given by ***ssitUserPathname*** using this base name. They will be ***filename***.dir and ***filename***.pag. For example, using **DATABASE=/home/jdoe/CERESchecklist** will result in two database files, **CERESchecklist.dir** and **CERESchecklist.pag**, being created in **/home/jdoe/**.
- The ***title*** is a user-selected name that will appear in the SSIT Manager GUI above the checklist items in the main window.

6 In the file, add the items that will appear in the checklist. Search for lines in the file of the form **ITEM=*ChecklistStep***. Each of these lines specifies a procedure or step taken during SSI&T. Edit these lines and/or add lines.

- The ***ChecklistStep*** is any text string (without quotes). Typically, each ***ChecklistStep*** is a SSI&T operational procedure or task. The source of these steps may be existing documentation, for example, the SSI&T Agreement between the Instrument Team and the DAAC.
- Add **ITEM** lines as necessary. Checklist items will appear in the SSIT Manager window in the order entered in this file.
- There is no limitation to the number of **ITEM** lines that can be entered. However, a large checklist will result in the SSIT Manager taking a significantly longer time to start up.

7 Save the changes made to the checklist file and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Enter**.
- For other editors, refer to that editor's documentation.

8 If database files already exist from a previous checklist set up, at the UNIX prompt on the AIT Sun, type **rm *ssitUserPathname/filename*.dir** press **Enter**. Type **rm s*sitUserPathname/filename*.pag**, press **Enter**.

- The ***ssitUserPathname*** and ***filename*** are the full path name and file name set in the checklist file in step 4.
- If these database files already exist (from a previous checklist set up), an error would result when the SSIT Manager was started.

9 At a UNIX prompt on the AIT Sun, type **vi ~/*mySSITpcf***, press **Enter**.

- The ***mySSITpcf*** is the file name of the private copy of the PCF used by the SSIT Manager (refer to Section 7.1, step 1).

10 In the file, search for identifier 603 beginning in the first column. The line should be of the form: **603|DpAtMgrLogDatabaseInit|*ssitUserPathname*/*filename***. Edit this line.

- The ***ssitUserPathname*** and ***filename*** are the full path name and file name set in the checklist file in step 4.

11 Save the changes made to the Process Control File and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Enter**.
- For other editors, refer to that editor's documentation.
- The set up of the SSIT Manager's checklist is now complete.

Running the SSIT Manager

This procedure describes the routine running of the SSIT Manager.

Running the SSIT Manager

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 **A**t the UNIX prompt on the AIT Sun from which the SSIT Manager is to be run, type **setenv DISPLAY *hostname*:0.0**, press **Enter**.

- The ***hostname*** is the name of the machine on which the SSIT Manager is to be displayed, *i.e.* the machine that your are using.
- If the machine your are on is an AIT Sun, this step should not be necessary; the variable should be set by default to **:0.0**, meaning the machine that you are on.
- As explained in Section 7.1, step 8, this command could be placed in the .cshrc file so that it is automatically set at login.
- To verify a setting, type **echo $DISPLAY**, press **Enter**.

4 If necessary, at the UNIX prompt on the AIT Sun from which the SSIT Manager is to be run, type **cleartool setview *ViewName***, press **Enter.**

- The ***ViewName*** is the ClearCase view to be used while the SSIT Manager is running in this session. For example, type **cleartool jdoe**, press **Enter**.
- A ClearCase view is required only if the SSIT Manager needs to be able to "see" into a ClearCase VOB; a view is not necessary otherwise.

5 At the UNIX prompt on the AIT Sun, type **DpAtMgr ConfigFile *ConfigFilename* &**, press **Enter**.

- The ***ConfigFilename*** is the configuration file name for the SSIT Manager.
- The **&** (ampersand) causes the SSIT Manager to be run as a background process, freeing up the UNIX prompt.

- Various messages from the SSIT Manager will appear in this window as it is running. For this reason, avoid using this window for other tasks until the SSIT Manager has terminated.

6 This step is optional. At the UNIX prompt on the AIT Sun, type **DpAtMgrLogDump *ssitUserPathname/filename***, press **Enter**.

- The ***ssitUserPathname*** and ***filename*** is the full path name and filename used in the **DATABASE=** line of the checklist file being used. See Section 7.2, step 4. For example, if the **DATABASE=** line in the checklist file was **DATABASE=/home/jdoe/CERESchecklist**, then type **DpAtMgrLogDump /home/jdoe/CERESchecklist**, press **Enter**.

- The contents of the database will be displayed on the screen. To redirect the output to a file, type **DpAtMgrLogDump *ssitUserPathname/filename* > *outputfile***, press **Enter**. The ***outputfile*** is the file name to be given to the output file.

The SSIT Manager window (Figure 1) includes two panes of information, a menu bar at the top of the window, and two buttons at the bottom of the window.

- **Checklist Pane -**The top pane is the SSI&T Checklist. This lists a configurable set of steps to be completed during the SSI&T process. It is displayed as a line of text for each item, with a check box that is checked or unchecked. Each checklist item has a unique ID. The checklist is manual, in that the operator must check it with a mouse click.
- **Log Pane -** The lower pane is a log of activities accomplished in the SSI&T process. As steps in the process are accomplished, a log entry is automatically added. Each time a box on the checklist is checked or unchecked, the log is updated. The user is required to annotate any changes.
- **Menu Bar -** The menu bar includes a simple selection of options:
    - Files - Offers the option to exit from the SSIT Manager application.
    - Tools - Provides access to a collection of tools that are of use in the SSI&T process. The tools that are available are described in the following section.
    - Run - User-customizable menu, intended for programs such as scripts to run science software.
- **Help -** Provides context-sensitive help.
- **SSIT Manager Buttons -** There are also two buttons at the bottom of the screen.
    - Edit - The checklist items may be edited by the SSI&T specialist to include additional detail on the SSI&T process. Clicking the EDIT button brings up the text editor for this purpose.

- Change State - This toggle button allows the SSI&T specialist to switch between “checked” and “unchecked” for the selected checklist item.

SSIT Manager Tools

There are several tools that are accessible through the SSIT Manager GUI. After selecting the TOOLS menu option of the menu bar, a set of options is available. See Figure 2 which indicates the use of the Tool menu item.

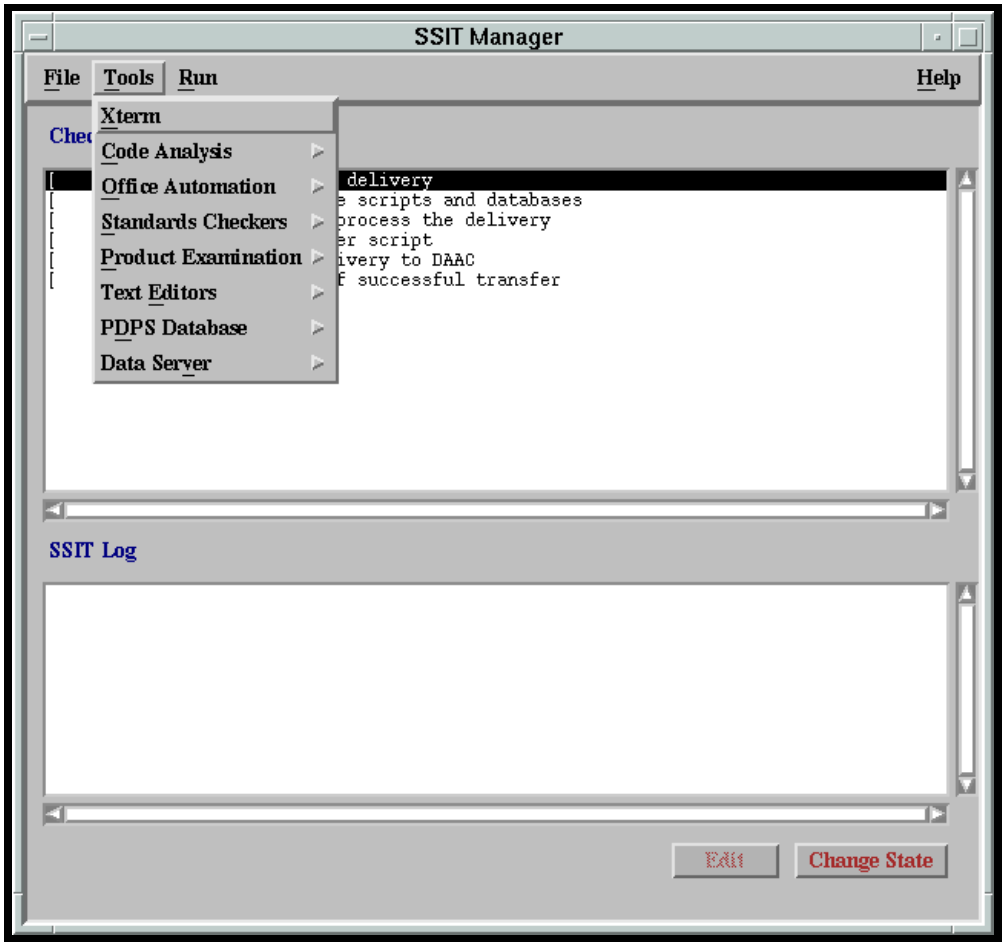

Figure 2. SSIT Manager Window - Tools Menu

- **Xterm -** will start an Xterm window session.
- **Code Analysis –** launches the SPARCworks application which is used to perform static code analysis (instrumental in detecting memory leaks.
- **Office Automation -** MS Windows, MS Office, Ghostview (used for displaying postscript format files), Netscape, Adobe Acrobat (displays PDF file format) and DDTS (tracking tool).
- **Standards Checkers -** provides access to the FORECHECK COTS tool or the Prohibited Function Checker custom application which verifies that science software follows coding standards established by the ESDIS Project. Additionally, the Process Control File (PCF) checker is provided to verify that the format of the PCF is correct.
- **Product Examination -** provides access to the Interactive Data Language (IDL) tool which is used to visualize and analyze scientific and engineering products, the EOSView tool which displays HDF format files and the metadata associated with those files, and the File Comparison tools which are provided to support file comparison of ASCII, Binary and HDF files.
- **Text Editors -** provides access to Emacs or Xedit tools which are editing tools used to develop scripts or ad hoc programs to be run in the SSIT environment.
- **PDPS Database -** provides access to the PCF ODL Template tool which creates a PDPS PGE science metadata ODL file from a science software PCF, the SSIT Science Metadata Update tool which updates PDPS/SSIT databases with PGE and ESDT science metadata read drom ODL files, and the SSIT Operational Metadata Update GUI tool which is used to update PDPS/SSIT database with PGE Operational metadata input by user.
- **Data Server -** provides access to the Copy SSIT to Production tool which is used to copy a single PGE version from the PDPS/SSIT database to the PDPS/Production database, the Register Subscription tool which is used to register a subscription for an ESDT with the Data Server, the Acquire DAP tool which is used to acquire a DAP from the Data Server, the Insert Static tool which is used to insert a static input file to the Data Server, the Insert Test Dynamic tool which is used to insert a test dynamic input file to the Data Server, the Insert EXE TAR tool which is used to insert a tar file of executables to the Data Server, the Insert SSAP tool which is used to insert a Science Software Archive Package (SSAP) to the Data Server and the Retrieve SSAP tool which is used to retrieve SSAP components from the Data Server.

Checklist for SSIT Manager

A checklist used to status software integration can be prepared to support the integration team. The checklist of operational procedures can be prepared in a UNIX session using any available editor such as Emacs, or Xedit. The following procedure may be used when creating a checklist through the SSIT Manager.

Preparing and Updating a Checklist

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 Click on the Checklist item to be modified and then select the **Change State** button.

- The text editor is launched.
- The checklist items may be edited.

5 Change the checklist items that need to be edited by typing the appropriate data in the Annotate Log Entry window.

6 Select the **OK** button.

- **Done** will be displayed in the bracketed column adjacent to the selected line of text.
- This must be manually completed after each listed item has been accomplished.
- The SSIT Log displays the new entry.

7 To quit this session, select the **File→Exit** from the menu bar.

This page intentionally left blank.

Standards Checking of Science Software

Standards Checking Overview

The purpose of standards checking is to verify that the source files of the science software are compliant with the ESDIS Data Production Software Computing Facility (SCF) Standards and Guidelines document.

Checking FORTRAN 77 ESDIS Standards Compliance

The ESDIS Data Production Software Computing Facility (SCF) Standards and Guidelines document requires all FORTRAN 77 code to be compliant with the ANSI FORTRAN 77. The COTS used for this task is FORCHECK.

Environment Variable:

- FORCHECK is run from the SUN that has the FORCHECK license.
- The Status Message Facility (SMF) files have been compiled.

FORCHECK can be started from the SSIT Manager but this procedure assumes that FORCHECK will be run from the command line using a script.

Checking FORTRAN 77 ESDIS Standards Compliance

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 At the UNIX prompt on the AIT Sun, type **vi *RunFile***, press **Enter**.

- The ***RunFile*** is the file name of the FORCHECK run script to be created. Any text editor may be used for this procedure step. The form of this script is shown in the template below:

```
forchk -l listfile.lst \
-I :incdir1:incdir2:[ … ] \
/pathname/source_code_1.f \
/pathname/source_code_2.f \
                .
```

.

.

/pathname/source_code_n.f

- The first line of ***RunFile*** should only contain the line: `forchk -l` ***ListFile*`.lst` **, as shown in the template above. The ***ListFile*** is the name to be given to the list output that will be produced. This file contains the source listing with messages produced by FORCHECK. Note that the terminal back slash (\) is required.
- The second line of ***RunFile*** contains the list of include (header file) directories. Each directory should be the full path name and each should be separated by a colon (:) only. A colon by itself should be placed at the beginning of the list (as shown in the template); this represents the current directory.
- Typical include directories to be used on the second line will be the include directories for the SDP Toolkit, HDF, and the science software itself.
- The second line should end in a terminal back slash (\).
- The third and subsequent lines should contain the full path names and file names of each FORTRAN 77 source file to be checked. Place each source file on a separate line. Terminate each line with a back slash (\) **except** the last line.
- Note that *all* lines in the ***RunFile*** end with a back slash (\) **except** the last one.
- Once completed save the file and exit the editor.

5 At the UNIX prompt on the AIT Sun, type **chmod +x *RunFile***, press **Enter**.

- The ***RunFile*** is the file name of the FORCHECK run script created in step 2.
- This command changes the file's disposition to "execute" allowing it to be run by invoking its name.

6 If required, at the UNIX prompt on the AIT Sun, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of a view allowing the FORTRAN 77 source files to be accessible.
- This step is only necessary if any of the FORTRAN 77 source files are in ClearCase (in the VOB under configuration management).

7 At the UNIX prompt on the AIT Sun, type ***RunFile* > & *FORCHECKoutput***, press **Enter**.

- The ***RunFile*** is the file name of the FORCHECK run script created in step 2.
- The ***FORCHECKoutput*** is the file name for the output file produced. Note that this output file differs from the ***ListFile*** specified in the FORCHECK run script.
- The **>&** is a C shell construct that causes standard error (where the output from FORCHECK normally emerges) to be redirected to a file.

8 At the UNIX prompt on the AIT Sun, type **vi *FORCHECKoutput***, press **Enter**.

- The ***FORCHECKoutput*** is the file name for the output file produced in step 5.
- The ***FORCHECKoutput*** file will contain any warnings, errors, and other messages from FORCHECK. A summary will be at the bottom of the file.
- Any text editor may be used for this procedure step.

9 At the UNIX prompt on the AIT Sun, type **vi *ListFile***, press **Enter**.

- The ***ListFile*** is the file name for the list file specified on line 1 in the FORCHECK run script.
- The ***ListFile*** file will contain FORCHECK messages similar to the ***FORCHECKoutput*** file embedded in the source code listing.
- Any text editor may be used for this procedure step.

Checking FORTRAN 90 Standards Compliance

The ESDIS Data Production Software Computing Facility (SCF) Standards and Guidelines document requires all FORTRAN 77 code to be compliant with the ANSI FORTRAN 90. The COTS used for this task is FORCHECK.

Environment Variable:

- The environment variable FCKCNF has be set to the location of the FORCHECK configuration file.
- FORCHECK is run from the SUN that has the FORCHECK license.
- The Status Message Facility (SMF) files have been compiled.

FORCHECK can be started from the SSIT Manager but this procedure assumes that FORCHECK will be run from the command line using a script.

Checking FORTRAN 90 ESDIS Standards Compliance

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SDPS SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Enter**. Then telnet to the SDPS SGI.

5 At the UNIX prompt on the SDPS SGI, type **setenv PGSHOME *ToolkitPathname***, press **Enter**. Then type, **source $PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Enter**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version (see Section 9.2).
- The ***sgiX*** refers to the appropriate processor (see Section 9.2). For example, type **source $PGSHOME/bin/sgi32/pgs-dev-env.csh**, press **Enter**.

6 If required, at the UNIX prompt on the SDPS SGI, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of a view allowing the Fortran 90 source files to be accessible.
- This step is only necessary if any of the Fortran 90 source files are in ClearCase (in the VOB under configuration management).

7 At the UNIX prompt on the SDPS SGI, type **cd *SrcPathname***, press **Enter**.

- The ***SrcPathname*** is the full path name to the location of the Fortran 90 source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB is the Fortran 90 source files are checked into ClearCase.

8 At the UNIX prompt on the SDPS SGI, type **f90 -c *[*-I$PGSINC*] [*-I$HDFINC*] [[*-IOtherIncFiles*…] SourceFiles* >& *ReportFile***, press **Enter**.

- The terms in square brackets (*[ ]*) are used to optionally specify locations of include and module (.mod) files. The **$PGSINC** already contains the SDP Toolkit include directory and **$HDFINC** already contains the HDF include directory. The **OtherIncFiles** represents one or more additional include or module directories.
- The ***SourceFiles*** is a list (space delimited) of Fortran 90 source files or a wildcard template (*e.g.* *.f90).
- The **>&** is a C shell construct that causes standard error (where the output from the Fortran 90 compiler normally emerges) to be redirected to a file.
- The ***ReportFile*** is the file name under which to save the results of the compile process.
- The **-c** flag causes only compilation (no linking).
- Apply the terms in square brackets only as necessary. Do not include the brackets in the actual command. See example below.
- Do not use the **-I** option for include or module files that are in the standard directories or in the current directory.
- The makefile for the science software may contain the names of additional include files needed by the software.
- For example, type **f90 -c -I$PGSINC -I$HDFINC -I/ecs/modis/pge5/include/*.f90 >& pge10.report**, press **Enter**.

9 At the UNIX prompt on the AIT Sun, type **vi *ReportFile***, press **Enter**.

- The ***ReportFile*** is the file name for the compilation results as produced in step 5.
- Any text editor may be used for this procedure step.

Checking C ESDIS Standards Compliance

The ESDIS Data Production Software Computing Facility (SCF) Standards and Guidelines document requires all C language source files are compliant with the ANSI.

The following is a list of tools, and or assumptions:

- The C source files are available in a single directory, accessible and have read permissions.
- Any non-standard include files are available in a single directory, accessible and have read permissions.
- The C compiler is accessible on the Science Processor (SGI Power Challenge).
- The current shell being used is the C shell (csh) or the T shell (tcsh).
- The Status Message Facility (SMF) files have been compiled.
- Compliance checking is done via an SGI Power Challenge C compiler flag.

Checking C ESDIS Standards Compliance

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SDPS SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Enter**. Then telnet to the SDPS SGI.

5 At the UNIX prompt on the SDPS SGI, type **setenv PGSHOME *ToolkitPathname***, press **Enter**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Enter**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version (see Section 9.2).
- The ***sgiX*** refers to the appropriate processor (see Section 9.2). For example, type **source $PGSHOME/bin/sgi32/pgs-dev-env.csh**, press **Enter**.

6 If required, at the UNIX prompt on the SDPS SGI, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of a view allowing the C source files to be accessible.

- This step is only necessary if any of the C source files are in ClearCase (in the VOB under configuration management).

7 At the UNIX prompt on the SDPS SGI, type **cd *SrcPathname***, press **Enter**.

- The ***SrcPathname*** is the full path name to the location of the C source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB is the C source files are checked into ClearCase.

8 At the UNIX prompt on the SDPS SGI, type **cc -c *[*-I$PGSINC*] [*-I$HDFINC*] [[*-IOtherIncFiles*…] SourceFiles* >& *ReportFile***, press **Enter**.

- The terms in square brackets (*[ ]*) are used to optionally specify locations of include and module (.mod) files. The **$PGSINC** already contains the SDP Toolkit include directory and **$HDFINC** already contains the HDF include directory. The **OtherIncFiles** represents one or more additional include directories.
- The ***SourceFiles*** is a list (space delimited) of C source files or a wildcard template (*e.g.* *.c).
- The **>&** is a C shell construct that causes standard error (where the output from the C compiler normally emerges) to be redirected to a file.
- The ***ReportFile*** is the file name under which to save the results of the compile process.
- The **-c** flag causes only compilation (no linking).
- Apply the terms in square brackets only as necessary. Do not include the brackets in the actual command. See example below.
- Do not use the **-I** option for include files that are in the standard directories (*e.g.* /usr/include) or in the current directory.
- The makefile for the science software may contain the names of additional include files needed by the software.
- For example, type **cc -c -I$PGSINC -I$HDFINC -I/ecs/modis/pge5/include/ *.c >& pge10.report**, press **Enter**.

9 At the UNIX prompt on the AIT Sun, type **vi *ReportFile***, press **Enter**.

- The ***ReportFile*** is the file name for the compilation results as produced in step 5.
- Any text editor may be used for this procedure step.

Checking Ada Standards Compliance

The ESDIS Data Production Software Computing Facility (SCF) Standards and Guidelines document requires that all Ada language source files be compliant with the ANSI.

The following is a list of tools, and or assumptions:

This procedure describes compiling Ada software using the COTS Verdix Ada Development System (VADS) which provides a complete environment for building (and developing) Ada software

Checking Ada Standards Compliance

---

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SDPS SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Enter**. Then telnet to the SDPS SGI.

5 If required, at the UNIX prompt on the SDPS SGI, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of a view allowing the Ada source files to be accessible.
- This step is only necessary if any of the Ada source files are in ClearCase (in the VOB under configuration management).

6 At the UNIX prompt on the SDPS SGI, type **setenv SGI_ABI -32**, press **Enter**.

- This command sets the environment variable **SGI_ABI** for 32-bit mode compilation.

7 At the UNIX prompt on the SDPS SGI, type **cd *SrcPathname***, press **Enter**.

- The ***SrcPathname*** is the full path name to the location of the Ada source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB if the Ada source files are checked into ClearCase.

8 At the UNIX prompt on the SDPS SGI, type **a.mklib**, press **Enter**.

- This command creates a VADS library directory. All Ada compilation must occur in a VADS Ada library.

9 At the UNIX prompt on the SDPS SGI, type **a.make -v -f *SourceFiles* >& *ReportFile***, press **Enter**.

- The ***SourceFiles*** is a list (space delimited) of Ada source files or a wildcard template (*e.g.* *.ada).
- The **>&** is a C shell construct that causes standard error (where the output from the Ada compiler normally emerges) to be redirected to a file.
- The ***ReportFile*** is the file name under which to save the results of the compile process.
- The **-v** flag enables verbose output.
- The **-f** flag indicates that what immediately follows are the source files. The order of the flags is therefore important.

10 At the UNIX prompt on the AIT Sun, type **vi *ReportFile***, press **Enter**.

- The ***ReportFile*** is the file name for the compilation results as produced in step 6.
- Any text editor may be used for this procedure step.

Prohibited Function Checker

The use of certain functions in the PGE is prohibited. The Prohibited Function Checker (Figure 4) is used to check C, FORTRAN 77, FORTRAN 90, and Ada language source files for the occurrence of functions that are prohibited in the ECS DAAC production environment.

The following is a list of tools, and or assumptions:

- The SSIT Manager is running.
- The source file(s) are available, accessible, and have read permissions.
- The below listed formatted text (ASCII) files containing the list of prohibited functions exist in the directory stored in the environment variable DPATMGR_DAT.
    - prohibitedFunctionsAda.txt
    - prohibitedFunctions.C++.txt
    - prohibitedFunctions.C.txt
    - prohibitedFunctions.F77.txt
    - prohibitedFunctions.F90.txt
- These files are installed with the SSIT Manager. They may be edited. Directions for editing are contained within each file.
- If the source code files to be checked are in a VOB in ClearCase™, a view has been set before the SSIT Manager was started.

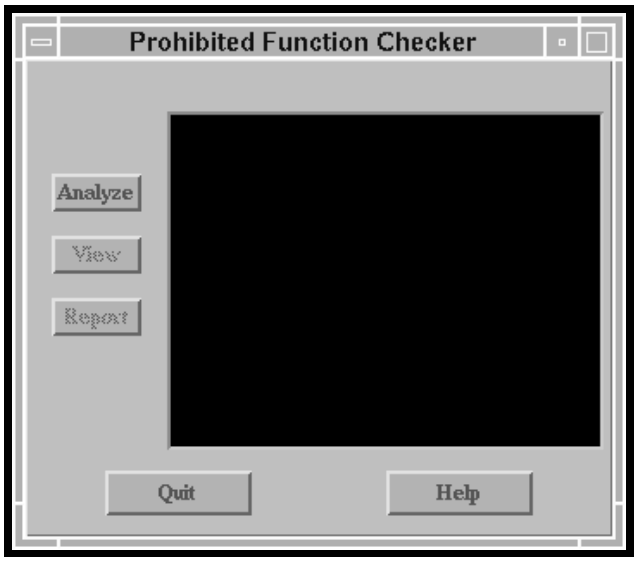

Figure 4. Prohibited Function

Prohibited Function Checker GUI

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Standards Checkers**. Then choose **Prohibited Function Checker**.

- The Prohibited Function Checker GUI will be displayed.

5 In the Prohibited Function Checker GUI, click on the **Analyze** button.

- The File Selector GUI will be displayed.

6 Within the **Directories** subwindow, double click on the desired directory.

- Repeat this step until the directory with the source files to be checked are displayed in the **Files** subwindow.

7 Within the **Files** subwindow, click on the source files to be checked. Each file clicked on will be highlighted.

- To choose groups of contiguous files, hold down the left mouse button and drag the mouse.
- To choose non-contiguous files, hold down the Control key while clicking on file names.

8 In the File Selector GUI, click on the **Ok** button.

- The File Selector GUI will disappear.
- The files selected in step 5 will be displayed in the Prohibited Function Checker GUI window as they are being checked.

9 In the Prohibited Function Checker GUI, click on the **Report** button.

- The **Report** GUI will be displayed.
- For each file, a list of prohibited functions found will be displayed.

10 Optionally, click on the **Print** button or the **Save** button.

- Choose **Save** to save the results to a file; choose **Print** to have the results printed on the default printer.
- Choosing **Save** will bring up a GUI labeled **Save To File**. Specify the directory and file name in which to save the results file.

11 Optionally, in the Prohibited Function Checker GUI, highlight one of the source files listed. Then click on **View**.

- The **Source Code** GUI will be displayed.
- Occurrences of prohibited functions found in that source file will be highlighted.
- Click on the **Next** button to bring into the window successive occurrences of prohibited functions (the **Next** button does not bring in the next source file).
- Click on the **Done** button to close the **Source Code** GUI. Other source files may be examined similarly, one at a time.

12 In the Prohibited Function Checker GUI, click on the **Quit** button.

- The Prohibited Function Checker GUI will disappear.
- This ends the session.

Prohibited Function Checker From the Command Line

---

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 If required, at the UNIX prompt on the AIT Sun, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of a view allowing the source files to be accessible.
- This step is only necessary if any of the source files are in ClearCase (in the VOB under configuration management).

4 At the UNIX prompt on the AIT Sun, type **cd *SrcPathname***, press **Enter**.

- The ***SrcPathname*** is the full path name to the location of the source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB if the source files are checked into ClearCase.
- The ***SrcPathname*** can contain other directories that contain source files and/or more directories. The Prohibited Function Checker will search out all source files in subdirectories recursively.

5 At the UNIX prompt on the AIT Sun, type **$DPATMGR_HOME/data/DPS/DpAtMgrBadFunc *fileorDirnames* > *ResultsFile***, press **Enter**.

- The ***fileorDirnames*** is a list of source file names or directory names of directories containing source files.
- The ***ResultsFile*** is the file name for the results that are output.

- For example, type **$DPATMGR_HOME/data/DPS/DpAtMgrBadFunc main.c utils/ > myOutput**, press **Enter**. Here, **main.c** is a source file and **utils/** is a directory that contains other source files.

6 At the UNIX prompt on the SDPS SGI, type **vi *ResultsFile***, press **Enter**.

- The ***ResultsFile*** is the file name for the output results as produced in step 3.
- Any text editor may be used for this procedure step.

Checking Process Control Files

The next task to accomplish is to check that the PCFs are syntactically correct and contain all necessary information for PGEs to run within the ECS DAAC production environment.

The following is a list of tools, and or assumptions:

- The SSIT Manager is running.
- The Process Control File(s) are available, accessible, and have read permissions.
- If the source code files to be checked are in a VOB in ClearCase™, a view has been set before the SSIT Manager was started.

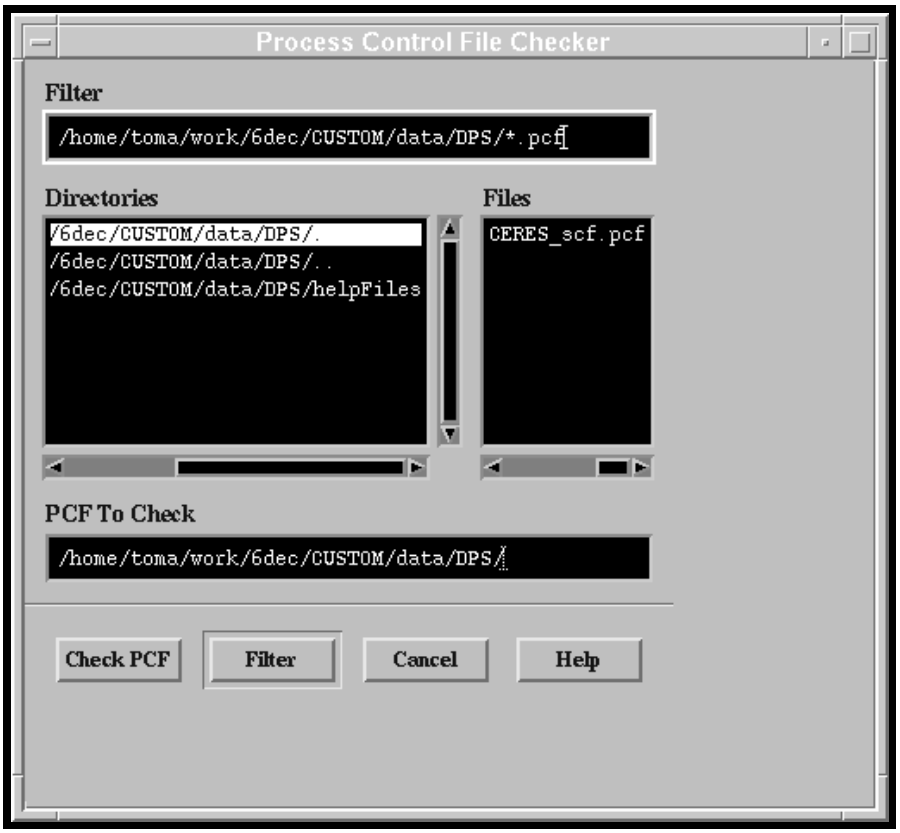

Figure 5. Process Control File Checker GUI

Checking Process Control Files: GUI Version

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Standards Checkers**. Then choose **Process Control File Checker**.

- The Process Control File Checker GUI will be displayed.

5 In the **Directories** subwindow, double click on the desired directory.

- Repeat this step until the directory with the PCF(s) to be checked are displayed in the Files window.
- Use the **Filter** subwindow to limit which files are displayed.

6 Within the **Files** subwindow, click on the PCF to be checked**.**

- The file clicked on will be highlighted.
- Only one PCF can be checked at a time.

7 Click on the **Check PCF** button.

- A GUI labeled **PCF Checker Results** will be displayed.
- Results will be displayed in this window.

8 Optionally, click on the **Save** button or on the **Print** button.

- Choose **Save** to save the results to a file; choose **Print** to have the results printed on the default printer.
- Choosing **Save** will bring up a GUI labeled **Save To File**. Specify the directory and file name in which to save the results file.
- Choosing **Print** and then clicking on the **OK** button will send the results to the default printer.

9 Click on the **Check Another** button or on the **Quit** button.

- Choosing **Check Another** allows another PCF to be checked. Repeat steps 2 through 5.
- Choosing **Quit** causes the Process Control File Checker GUI to disappear and ends the session.

Checking Process Control Files: Command Line Version

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 If not already done, at the UNIX prompt on the SDPS SGI, type **setenv PGSHOME *ToolkitPathname***, press **Enter**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Enter**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version (see Section 9.2).
- The ***sgiX*** refers to the appropriate processor (see Section 9.2). For example, type **source $PGSHOME/bin/sgi32/pgs-dev-env.csh**, press **Enter**.
- For checking Process Control Files, the choice of which SDP Toolkit version is set up is irrelevant.

4 If required, at the UNIX prompt on the SDPS SGI, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of a view allowing the Process Control File(s) to be accessible.
- This step is only necessary if any of the Process Control Files are in ClearCase (in the VOB under configuration management).

5 At the UNIX prompt on the SDPS SGI, type **cd *PCFpathname***, press **Enter**.

- The ***PCFpathname*** is the full path name to the location of the Process Control File(s) to be checked.
- The ***PCFpathname*** will be in the ClearCase VOB if the Process Control Files are checked into ClearCase.

6 At the UNIX prompt on the SDPS SGI, type **$PGSBIN/pccheck.sh -i *PCFfilename* > *ResultsFile***, press **Enter**.

- The ***PCFfilename*** is the full path name (directory and file name) to the Process Control File to check.
- The ***ResultsFile*** is the file name for the results that are output.

7 At the UNIX prompt on the SDPS SGI, type **vi *ResultsFile***, press **Enter**.

- The ***ResultsFile*** is the file name for the output results as produced in step 4.
- Any text editor may be used for this procedure step.

Extracting Prologs

The Project standards and guidelines are contained in the document *Data Production Software and Science Computing Facility (SCF) Standards and Guidelines, Revision A, October 1996* (423-16-01). This ESDIS document mandates that science software delivered to the DAACs to be integrated into the ECS contain prologs in the source files. Prologs are internal documentation containing information about the software. The details are specified in the ESDIS document. Prologs must be at the top of every function, subroutine, procedure, or program module.

This procedure describes using the Prolog Extractor to extract prologs into a file. Note that the prolog extractor only extract the prologs it finds. It does not check the contents of prologs.

The procedures assume that the SSIT Manager is running and that the source files from which prologs are to be extracted are available, accessible, and have read permissions for the operator.

Prologs are assumed to be delimited by particular delimiters depending on the language type. Delimiters are listed in the table below:

Prolog Delimiters

Language	Type	Delimiter
FORTRAN 77	source	!F77
Fortran 90	source	!F90
C	source	!C
Ada	source	!Ada
FORTRAN 77	include	!F77-INC
Fortran 90	include	!F90-INC
C	include	!C-INC
Any Language	any	!PROLOG
All Languages	The end delimiter is always !END	

The Prolog Extractor recognizes the language type of the file by its file name extension. The table below lists assumed file name extensions:

File Name Extensions

File Type	File Name Extensions
FORTRAN 77	f, f77, ftn, for, F, F77, FTN, FOR
Fortran 90	f90, F90, f, F
FORTRAN 77/Fortran 90 include	inc, INC
C	c
C/C++ header	h
Ada	a, ada

Checking Process Control Files: Command Line Version

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 From the SSIT Manager, click on the **Tools** menu, then choose **Standards Checkers**. Then choose **Prolog Extractor** .

- An xterm (on the AIT Sun) will be displayed within which the Prolog Extractor will be run.
- The Prolog Extractor can also be started from the UNIX prompt. To do this, at the UNIX prompt on the AIT Sun, type **DpAtMgrPrologs**, press **Enter**.

4 At the **Files(s)? (-h help)** prompt, type in file names and/or directory names containing the files.

- Separate items with spaces.
- If item is a directory (and it exists within the current directory), the contents of the directory will be search recursively for files with valid file name extensions (see Table 8.7-3).
- Use **./** to indicate the current directory. The extractor will search for all files in the current directory and recursively below.
- The current directory is the directory from which the SSIT Manager was started.
- The time needed for the Prolog Extractor could be very long for large numbers of files and directories.

Compiling and Linking Science Software

After the files have been checked, they must be prepared for the Science Data Processing Segment (SDPS) environment. The next step in this process is to resolve any conflicts that the PCF Checker found, compile status message facility files, set up an SCF version SDP Toolkit environment, compile a PGE and link with the SCF version of the SDP toolkit.

Updating the Process Control File

Once the PCF has been checked, these files may require updating. The Process Control File consist of the following sections.

- System Runtime Parameters
- Product Input
- Product Output
- Support Input
- Support Output
- User-defined Runtime Parameters
- Intermediate Input
- Intermediate Output
- Temporary IO

Each of these sections may require updating preparation for the SDPS environment.

Updating the Process Control Files (PCFs)

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 Select the **Tools→Standards Checker→ PCF Checker** from the SSIT Manager menu bar.

- The PCF Checker GUI will be displayed.

5 Find the **Directories** subwindow and double-click on the directory containing the PCF(s).

- The files contained within the directory will be displayed in the **Files** subwindow.

6 Within the **Files** subwindow, highlight the PCF file to check and click on the **Check PCF** button.

- The PCF Checker Results GUI will be displayed. Errors and warnings found will be listed in the main subwindow.

7 If any errors are detected, fix them.

- This ensures that the PCF is syntactically correct before it is modified and contains no detectable errors.

8 Select the **Tools→Xterm** from the SSIT Manager menu bar.

- An Xterm window is displayed.

9 If the PCF has been checked into ClearCase™, then check it out by typing**: cleartool checkout -nc PCFfilename** (where **PCFfilename** is the filename of the PCF.

10 Where necessary, make changes to the default directories specified in each section of the PCF.

- Each section begins with a line consisting of a **?.**
- Each of these section heading lines will be followed by a line that begins with a **!** .
- If a line reads: **! ~ / runtime** leave it unchanged. The (~) is a symbol for PGSHOME.
- When specifying a default directory other than **~/runtime,** do not use a tilde.

11 Look for science software specific entries in each section and make changes to paths (field 3) as necessary

- All paths specified in the PCF must exist on the SGI Power Challenge at the DAAC.
- The science software specific entries will have logical Ids outside the range 10,000-10999 (in the first field).

12 Add the necessary entry to the SUPPORT OUTPUT FILES section by typing: **10113 | eventLogger.log | | | | |** .

- Type entry on the same line, exactly as shown.

13 Verify that the SUPPORT OUTPUT FILES section contains the following entry to the shared memory pointer file:

10111 | ShmMem | ~/runtime | | | |1

If this line is not within this section, add it to the SUPPORT OUTPUT FILES section.

14 Verify that the USER DEFINED RUNTIME PARAMETERS section contains the following entry:

10112 | comment | 1

If this line is not within this section, add it to the USER DEFINED RUNTIME PARAMETERS section.

15 Verify (and change if necessary) that the location of certain database files pointed to in the PCF read as follows:

10301 |leapsec.dat |~/lib/database/TD | | | |1

10401 | utcpole.data |~/lib/database/CSC | | | |1

10402 | earthfigure.dat |~/lib/database/CSC | | | |1

10501 | . |~/lib/database/EPH | | | |1

10601 | de200.eos |~/lib/database/CBP | | | |1

Change the **~/lib/database** part of each path to **~/ database /mode**

16 Re-invoke the Process Control File Checker by selecting **Tools→Standards Checker→ PCF Checker** from the SSIT Manager menu bar. If any errors are detected, fix them.

17 If the PCF had been checked out of ClearCase™, check the revised version back in by clicking on the Xterm window and typing cleartool checkin -nc **PCFfilename** (where **PCFfilename** is the filename of the PCF.

Setting up a SDP Toolkit Environment

The purpose of the SDP Toolkit is to allow science software to be developed for ECS at independent SCFs and to provide:

- An interface to the ECS system, including PDPS and CSMS and information management.
- A method for Science software to be portable to different platforms at the DAAC.
- A method to reduces redundant coding at the SCF.
- Value added functionality for science software development.

The SDP Toolkit is divided into two groups of tools:

Mandatory Tools

- Error and Status Message Facility (SMF) - provides general error handling, status log messaging, and interface to CSMS services.
- Process Control Tools - provides the primary interface to the PDPS. Allows access to physical filenames and file attributes and retrieval of user defined parameters.
- Generic Input/Output - provides the means to open and close support, temporary and intermediate duration files.
- Memory Allocation Tools - simple wrappers on native C functions which track memory usage in the SDPS, and shared memory tools which enable the sharing of memory among executables within a PGE.

Optional Tools

- Ancillary Data Access - provides access to NMC data and Digital Elevation (DEM) data.
- Celestial Body Position - locates the sun, moon and the planets.
- Coordinate System Conversion - coordinate conversions between celestial reference.
- Constant and Unit Conversion - physical constants and unit conversions.
- IMSL - mathematical and statistical support.

In the description of the Toolkit routines, descriptive information is presented in the following format:

TOOL TITLE

NAME:	Procedure or routine name
SYNOPSIS:	C: C language call
FORTRAN:	FORTRAN77 or FORTRAN90 language call
DESCRIPTION:	Cursory description of routine usage
INPUTS:	List and description of data files and parameters input to the routine
OUTPUTS:	List and description of data files and parameters output from the routine
ENTERS:	List of returned parameters indicating success, failure, etc.
EXAMPLES:	Example usage of routine
NOTES:	Detailed information about usage and assumptions
REQUIREMENTS:	Requirements from PGS Toolkit Specification, Oct. 93 which the routine satisfies

The science software delivered to the DAACs is expected to work with either the SCF SDP Toolkit or the DAAC SDP Toolkit which are both installed each DAAC. During the pre-SSI&T initial testing, the SCF Toolkit should be used.

There are several versions of the SCF/DAAC SDP Toolkit installed on the SGI Power Challenges at the DAACs for the Testbed. The toolkit versions at the DAACs differ according to:

- Object Type - The operating system on the SGI Power Challenges on the Testbed is IRIX 6.2, a 64-bit operating system. To be backward compatible, the SGI operating system will allow new 64-bit and 32-bit objects to be built as well as the older 32-bit machines. Each of these object types are designated by placing a cc flag on the command line to enable a particular mode with the SGI C compiler.
  - New 64-bit: cc flag = -64
  - New 32-bit: cc flag = -n32
  - Old 32-bit: cc flag = -32
- Library Type - The SDP Toolkit uses different libraries depending upon whether FORTRAN 77 or FORTRAN 90 source code is being linked. If C source code is to be linked, then either language version of the library will work.

The following Table summarizes the available SDP Toolkits used by the SGI science processors.

SDP Version	OS Compatibility	Language Type	Library Object Type	PGSHOME
SCF	IRIX 5.3	FORTRAN 77 or C	old 32-bit mode	/RelA_IT/scf_toolkit_f77/TOOLKIT
SCF	IRIX 5.3	Fortran 90 or C	old 32-bit mode	/RelA_IT/scf_toolkit_f90/TOOLKIT
DAAC	IRIX 5.3 & 6.1	FORTRAN 77 or C	old 32-bit mode	/RelA_IT/daac_toolkit_f77/TOOLKIT
DAAC	IRIX 5.3 & 6.1	Fortran 90 or C	old 32-bit mode	/RelA_IT/daac_toolkit_f90/TOOLKIT
DAAC	IRIX 6.1	FORTRAN 77 or C	new 32-bit mode	/RelA_IT/daac_toolkit_f77/TOOLKIT
DAAC	IRIX 6.1	Fortran 90 or C	new 32-bit mode	/RelA_IT/daac_toolkit_f90/TOOLKIT
DAAC	IRIX 6.1	FORTRAN 77 or C	new 64-bit mode	/RelA_IT/daac_toolkit_f77/TOOLKIT
DAAC	IRIX 6.1	Fortran 90 or C	new 64-bit mode	/RelA_IT/daac_toolkit_f90/TOOLKIT

SDP Version - indicates whether the SDP Toolkit is a DAAC or SCF version.

OS Compatibility - indicates under which operating system the Toolkit was compiled.

Language Type - indicates which languages the Toolkit is to be used.

Library Object Type - indicates which mode the Toolkit library was compiled.

PGSHOME - indicates the location of the SDP Toolkit home directory for that version.

There are eight environments available. Setting up a SDP Toolkit environment means defining a number of environment variables to point to various locations within the particular SDP Toolkit Hierarchy. By using these environment variables, build scripts or make files can be written to be independent of a particular SDP Toolkit version.

PGSHOME - The environment variable PGSHOME must be set on the science processor SGI to the home directory of the appropriate SDP home directory.

> Input using C Shell by typing: **setenv PGSHOME sdp-home-dir** where **sdp-home-dir** is the home directory of the desired SDP Toolkit version.

Other Environment Variables - Additional SDP Toolkit environment variables may be set as follows:

> Input using C Shell by typing: **source $PGSHOME/bin/sgiX/pgs-dev-env.csh** where sgiX is replaced with sgi for old 32-bit version of the SDP Toolkit, sgi32 for new 32-bit version of the SDP Toolkit or sgi64 for new 64-bit version of the SDP Toolkit.

Using the table listed above, these environment variables would be similar to the following example:

> **seten PGSHOME /vola1/RelA/DAAC_toolkit_f77/TOOLKIT**
>
> **source $PGSHOME/bin/sgi32/pgs-dev-env.csh**

Setting Up the SDP Toolkit Environment

---

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the SDPS SGI, type **setenv PGSHOME *ToolkitPathname***, press **Enter**.

- The ***ToolkitPathname*** is the home directory of the particular SDP Toolkit version being used. Refer to Table 9.2-2. Note that the setting of PGSHOME shown in this table may differ in your local DAAC.

- Korn shell users, type **PGSHOME=*ToolkitPathname*; export PGSHOME**, press **Enter**.

4 At the UNIX prompt on the SDPS SGI, type **source $PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Enter**.

- The ***sgiX*** is one of: **sgi32** for 32-bit version of the Toolkit or **sgi64** for 64-bit version of the Toolkit.
- Korn shell users, type **. $PGSHOME/bin/*sgiX*/pgs-dev-env.ksh**, press **Enter** (note the "dot" at the beginning of this command).

5 This step is optional. Edit the file $HOME/.cshrc and add the line **alias *aliasname* 'setenv PGSHOME *ToolkitPathname*; source $PGSHOME/bin/*sgiX*/pgs-dev-env.csh; echo "*textmessage*" '**.

- The ***aliasname*** is the name of the alias. For example, to set up an environment for the DAAC version of the Toolkit for FORTRAN 77 (or C), you might use **DAACf77** as an ***aliasname***.
- The ***ToolkitPathname*** is the home directory of the particular SDP Toolkit version being used. Refer to Table 9.2-2. Note that the setting of PGSHOME shown in this table may differ in your local DAAC.
- The ***sgiX*** is one of: **sgi32** for 32-bit version of the Toolkit or **sgi64** for 64-bit version of the Toolkit.
- The ***textmessage*** is a message that will be echoed to the screen signifying that a new Toolkit environment has been set up. It must be enclosed within double quotes ("). An example may be, **"DAAC F77 Toolkit environment is now set."**
- A complete example (it should be all on one line in the .cshrc file):

  alias DAACf77 'setenv PGSHOME /RelA/daac_toolkit_f77/TOOLKIT; source $PGSHOME/bin/sgi32/pgs-dev-env.csh; echo "DAAC F77 Toolkit environment is now set" '

- Other aliases for other versions of the Toolkit can be set up similarly.

Linking a PGE with SCF Version of SDP Toolkit

The linking of science software will vary according to the particular delivery. This procedure will follow some of the "typical" steps that may or may not be applicable to a particular situation.

The following is a list of tools, and or assumptions:

- All instructional information supplied with the delivery should be read prior to compiling and linking a PGE with the SCF version of the SDP Toolkit. These instructions should be the primary source of information.
- The science software delivery has been unpacked and placed into the software build area.
- All files necessary to build the science software are available, accessible, and have the proper permissions set.
- The build process will be done on the SGI Power Challenge in a UNIX shell.

Linking a PGE with SCF Version of SDP Toolkit

---

1 Read all instructional material supplied with the science software delivery. Such material should be the primary source of information on how to build the science software.

- Read the *Systems Description* document and the *Operations Manual.* Both of these or their equivalent should be in the delivery.
- Typically, there will be "readme" files accompanying each PGE in the directory structure, perhaps in a doc directory.
- Text files (ASCII) may be viewed with the UNIX command, *more* or with the *vi* editor.
- PostScript documents may be viewed with *ghostview,* which is accessible via the SSIT Manager.
- PDF formatted documents may be viewed with *acroread*, the Acrobat Reader, also accessible via the SSIT Manager.
- Documents in Microsoft Word and related formats may be viewed through the Microsoft Windows™ 3.1 emulator. The MS Windows emulator may be accessed from the SSIT Manager.

2 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter.**

3 Enter the **password** then press **Enter.**

4 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SDPS SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Enter**. Then telnet to the SDPS SGI.
- It is recommended that this procedure begin within a new command shell on the SDPS SGI.

5 At the UNIX prompt on the SDPS SGI, type **setenv PGSHOME *ToolkitPathname***, press **Enter**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Enter**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, an SCF version (see Section 9.2).
- The ***sgiX*** refers to the appropriate processor (see Section 9.2). For example, type **source $PGSHOME/bin/sgi32/pgs-dev-env.csh**, press **Enter**.

6 If make files are in ClearCase, at the UNIX prompt on the SDPS SGI, type **cleartool setview *ViewName***, press **Enter**. Then, **cd *pathname***, press **Enter**. And **cleartool checkout -nc *makefile***, press **Enter**.

- The ***ViewName*** is the name of a view allowing the make files to be accessible.
- The ***pathname*** is the full path name of the directory (in the VOB) where the make file has been checked in.
- The ***makefile*** is the name of the make file to examine and possibly modify.
- This step is only necessary if any of the make files (or build scripts) are in ClearCase (in the VOB under configuration management).

7 Examine and alter (if necessary) any make files using any text editor (*vi, emacs*).

- There may be several make files for a particular PGE.
- Verify that compiler, compiler flag settings, and other environment variable settings are appropriate.
- The Toolkit set up (from step 3) will set many environment variables which can be used in the make files. To see the current environment variable settings, at the UNIX prompt on the SDPS SGI, type **env**, press **Enter**.

8 Compile any required status message facility (SMF) files and place the header file(s) in the proper directory for building. See Section 9.3.

9 Verify that the directory structure for the PGE source files matches the directory structure expected by the make files or build scripts.

- Deliveries may come with install scripts that place files into various directories according to some predefined structure.

10 If necessary, at the UNIX prompt on the SDPS SGI, type **cleartool checkout -nc *filename***, press **Enter**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked out of ClearCase. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check out step.
- Note that checking in executable or object files is *not* recommended in the first place.

11 Build the software in accordance with instructions delivered.

- Science software deliveries may come with a single, top-level script to do the entire build or the build process could involve a series of steps, each of which should be described fully in the delivered documentation.

12 If necessary, at the UNIX prompt on the SDPS SGI, type **cleartool checkin *filename* -nc**, press **Enter**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked into ClearCase. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check in step.
- Note that checking in executable or object files is *not* recommended.

Linking a PGE with the DAAC Version of the SDP Toolkit

In order to be tested at the DAAC, science software must be compiled and linked to produce binary executables. These binary executables are then packaged into one or more shell scripts as defined by the science software developer (Instrument Team). These science software packages are the Product Generation Executives (PGEs) delivered to the DAACs during SSI&T. PGEs are the smallest schedulable unit of science software in the ECS.

Building science software into PGEs should be done in accordance with supplied documentation. Such documentation should describe the process in detail. In general, science software deliveries will come with make files or other build scripts to automate the build process.

In general, science software will be built, run, and tested with the SCF version of the SDP Toolkit to ensure that the software has been successfully ported to the DAAC. Once this test has been completed successfully, the science software will be re-built, rerun, and re-tested with the DAAC version of the SDP Toolkit. Only with the DAAC Toolkit can the PGE be run within the ECS.

This procedure describes some general principals that may or may not be applicable to a particular science software delivery for building a PGE with the DAAC version of the SDP Toolkit. See Section 9.4 for building a PGE with the SCF version of the SDP Toolkit.

The procedures assume that the C shell (or a derivative) is the current command shell.

Linking a PGE with DAAC Version of the SDP Toolkit

1 Read all instructional material supplied with the science software delivery. Such material should be the primary source of information on how to build the science software.

- Read the *Systems Description* document and the *Operations Manual.* Both of these or their equivalent should be in the delivery.
- Typically, there will be "readme" files accompanying each PGE in the directory structure, perhaps in a doc directory.
- Text files (ASCII) may be viewed with the UNIX command, *more* or with the *vi* editor.
- PostScript documents may be viewed with *ghostview,* which is accessible via the SSIT Manager.
- PDF formatted documents may be viewed with *acroread*, the Acrobat Reader, also accessible via the SSIT Manager.
- Documents in Microsoft Word and related formats may be viewed through the Microsoft Windows™ 3.1 emulator. The MS Windows emulator may be accessed from the SSIT Manager.

2 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SDPS SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Enter**. Then telnet to the SDPS SGI.
- It is recommended that this procedure begin within a new command shell on the SDPS SGI.

3 At the UNIX prompt on the SDPS SGI, type **setenv PGSHOME *ToolkitPathname***, press **Enter**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Enter**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, a DAAC version (see Section 9.2).
- The ***sgiX*** refers to the appropriate processor (see Section 9.2). For example, type **source $PGSHOME/bin/sgi32/pgs-dev-env.csh**, press **Enter**.

4 If make files are in ClearCase, at the UNIX prompt on the SDPS SGI, type **cleartool setview *ViewName***, press **Enter**. Then, **cd *pathname***, press **Enter**. And **cleartool checkout -nc *makefile***, press **Enter**.

- The ***ViewName*** is the name of a view allowing the make files to be accessible.
- The ***pathname*** is the full path name of the directory (in the VOB) where the make file has been checked in.
- The ***makefile*** is the name of the make file to examine and possibly modify.

- This step is only necessary if any of the make files (or build scripts) are in ClearCase (in the VOB under configuration management).

5 Examine and alter (if necessary) any make files using any text editor (*vi, emacs*). If the software had already been built and tested with the SCF version of the SDP Toolkit, this step may be unnecessary.

- There may be several make files for a particular PGE.
- Verify that compiler, compiler flag settings, and other environment variable settings are appropriate.
- The Toolkit set up (from step 3) will set many environment variables which can be used in the make files. To see the current environment variable settings, at the UNIX prompt on the SDPS SGI, type **env**, press **Enter**.

6 Compile any required status message facility (SMF) files and place the header file(s) in the proper directory for building. See Section 9.3.

7 Verify that the directory structure for the PGE source files matches the directory structure expected by the make files or build scripts.

- Deliveries may come with install scripts that place files into various directories according to some predefined structure.

8 If necessary, at the UNIX prompt on the SDPS SGI, type **cleartool checkout -nc *filename***, press **Enter**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked out of ClearCase. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check out step.
- Note that checking in executable or object files is *not* recommended in the first place.

9 Build the software in accordance with instructions delivered.

- Science software deliveries may come with a single, top-level script to do the entire build or the build process could involve a series of steps, each of which should be described fully in the delivered documentation.

10 If necessary, at the UNIX prompt on the SDPS SGI, type **cleartool checkin *filename* -nc**, press **Enter**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked into ClearCase. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check in step.
- Note that checking in executable or object files is *not* recommended.

Linking the PGE to the Status Message Facility File (SMF)

The SMF is also known as the Error Status Message. Its purpose is to provide an error and status message handling mechanism for use in science software, and to provide a method to send log files, informational messages and output data files to DAAC personnel or to remote users.

The SCF SDP Toolkit makes extensive use of PGS_SMF functions for error checking purposes. Much effort has gone into assuring that the maximum number of possible errors will be trapped, without sacrificing the speed and efficiency of the Toolkit code.

The SMF files need to be linked with the science software into message files and include (header) files. These files will be used by the science software during runtime.

The following is a list of tools, and or assumptions:

- The delivered SMFs are available, accessible, and have read permissions.
- The delivered smfcompile is available.

The following procedure will review the process required to compile the SMF files with the science software.

Compiling SMF Files

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SDPS SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Enter**. Then telnet to the SDPS SGI.
- It is recommended that this procedure begin within a new command shell on the SDPS SGI.

5 If required, at the UNIX prompt on the SDPS SGI, type **cleartool setview *ViewName***, press **Enter**.

- The ***ViewName*** is the name of a view allowing the SMF files to be accessible.
- This step is only necessary if any of the SMF files are in ClearCase (in the VOB under configuration management).

6 At the UNIX prompt on the SDPS SGI, type **setenv PGSHOME *ToolkitPathname***, press **Enter**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Enter**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version (see Section 9.2).
- The ***sgiX*** refers to the appropriate processor (see Section 9.2). For example, type **source $PGSHOME/bin/sgi32/pgs-dev-env.csh**, press **Enter**.

7 At the UNIX prompt on the SDPS SGI, type **cd *pathname***, press **Enter**.

- The ***pathname*** is the full path name to the directory containing the SMF text files.
- The SMF text files will typically have .t file name extensions.

8 At the UNIX prompt on the SDPS SGI, type **smfcompile -*lang* -f *textfile*.t**, press **Enter**.

- The **-*lang*** is a flag that indicates for what language to compile. This flag can be one of **-c** to produce C header files, **-f77** to produce FORTRAN 77 include files, and **-ada** to produce Ada include files. The default is for C include files. For example, type **smfcompile -f77 PGS_MODIS_39123.t**, press **Enter**.
- The ***textfile*** is the file name of the SMF text file delivered with the science software.
- The SMF text files will typically have .t file name extensions.
- File names for SMF text files usually have the "seed" value used by the file as part of its file name (*e.g.* PGS_MODIS_39123.t where 39123 is the seed number).
- Only one such SMF text file can be compiled at a time; wildcards cannot be used.
- The SMF compiler may be run with the additional flags **-r** and **-i** as in, **smfcompile -f *textfile*.t -r -i**. The **-r** automatically places the runtime message file in the directory given by the environment variable PGSMSG. The **-i** automatically places the include file in the directory given by the environment variable PGSINC. For example, type **smfcompile -ada -r -i -f PGS_MODIS_39123.t**, press **Enter**. Note that the **-f** flag must always be immediately followed by the name of the text file.

9 If necessary, at the UNIX prompt on the SDPS SGI, type **mv *IncludeFilename* $PGSINC**, press **Enter**. Then, type **mv *RuntimeFilename* $PGSMSG**, press **Enter**.

- This step is only required if either the **-r** or the **-i** flag were not used in step 5.
- The ***IncludeFilename*** is the name of the include file created in step 5.
- The ***RuntimeFilename*** is the name of the runtime message file created in step 5.

- For example, type **mv PGS_MODIS_39123.h $PGSINC**, press **Enter**. And then type, **mv PGS_MODIS_39123 $PGSMSG**, press **Enter**.

This page intentionally left blank.

Running a PGE in a Simulated SCF Environment

Science software delivered to the DAACs for SSI&T was developed and tested at individual SCFs using the SCF version of the SDP Toolkit. Before linking the software with the DAAC version of the Toolkit and integrating it with the ECS, it is prudent to first link the software to the SCF version of the Toolkit and run it as was run at the SCF. This type of testing can reveal problems associated with the process of porting the software to another platform whose architecture may be quite different from the one on which the software was developed.

A simulated SCF environment means that the software is built using the SCF version of the Toolkit and is run from the UNIX command line. The Planning and Data Processing System (PDPS) and the IMF Data Server are not involved.

The procedures which follow describe how to run the science software in a simulated SCF environment.

Setting Up the Environment for Running the PGE

Running a PGE that has been built with the SCF version of the SDP Toolkit requires some environment set up as it does at the SCF. This procedure describes how to set up a simulated SCF environment.

The procedures assume that the Process Control File (PCF) exists and has been tailored for the DAAC environment (Section 9.1) and that the C shell or a derivative (*e.g.* T shell) is the current user shell.

Setting Up the Environment for Running the PGE

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SDPS SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Enter**. Then telnet to the SDPS SGI.

- It is recommended that this procedure begin within a new command shell on the SDPS SGI.

5 At the UNIX prompt on the SDPS SGI, type **setenv PGSHOME *ToolkitPathname***, press **Enter**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Enter**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, an SCF version (see Section 9.2).
- The ***sgiX*** refers to the appropriate processor (see Section 9.2). For example, type **source $PGSHOME/bin/sgi32/pgs-dev-env.csh**, press **Enter**.

6 At the UNIX prompt on the SDPS SGI, type **setenv PGS_PC_INFO_FILE *PCFpathname*/*PCFfilename***, press **Enter**.

- The ***PCFpathname*** is the full path name to the location of the Process Control File (PCF) to be associated with this PGE.
- The ***PCFfilename*** is the file name of the PCF.
- For example, **setenv PGS_PC_INFO_FILE /disk2/PGE32/PCF/PGE32.pcf**, press **Enter**.

7 Optionally, at the UNIX prompt on the SDPS SGI, type **rm *LogPathname*/*LogFilename***, press **Enter**.

- The ***LogPathname*** is the full path name to the location of the PGE log files for this PGE.
- The ***LogFilename*** is the file name of the PGE log file to remove from a previous run of the same PGE. PGE log files can be Status, User, or Report.
- The ***LogFilename*** may use wildcard characters to remove all of the log files at the same time.
- This step is optional. If log files from a previous run of the same PGE are not removed, they will be appended with the information from the current run.
- The environment will then be set up. Continue on to Section 10.2.

Running and Profiling the PGE

Profiling a PGE refers to the process of gathering information about the runtime behavior of a PGE. The information includes the wall clock time, user time and system time devoted to the PGE; the amount of memory used; the number of page faults; and the number of input and output blocks.

The Planning and Data Processing System (PDPS) database must be populated with the above information when the PGE is registered with the PDPS during the integration phase of SSI&T. This information may be delivered with the PGE or it may need to be determined at the DAAC during SSI&T. This procedure addresses the latter need.

Note that profiling, as used here, does not involve altering the binary executable to produce instrumented code.

The procedures assume that the PGE has been built successfully with the SCF version of the SDP Toolkit (Section 9.4), the required SMF runtime message files have been produced and placed in the correct locations (Section 9.2), that the Process Control File (PCF) exists and has been tailored for the DAAC environment (Section 9.1), and that the required environment for running the PGE has been set up (Section 10.1). It is further assumed that the required input files are available and accessible and that the C shell or a derivative (*e.g.* T shell) is the current user shell.

Running and Profiling the PGE

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the SDPS SGI in the window containing the set up environment from Section 10.1, type **cd *PGEbinPathname***, press **Enter**.

- The ***PGEbinPathname*** is the full path name of the directory containing the built PGE binary executable. For example, **cd /disk2/PGE32/bin/**, press **Enter**.The ***PGEbinPathname*** is the full path name of the directory containing the built PGE binary executable. For example, **cd /disk2/PGE32/bin/**, press **Enter**.

4 At the UNIX prompt on the SDPS SGI, type ***DpPrR*usage *PGE* >& *ResultsOut***, press **Enter**.

- The ***PGE*** is the name given to the PGE binary executable.
- The ***ResultsOut*** is the file name in which to capture the profiling results as well as any messages from standard output (stdout) and standard error (stderr) that may be produced by the running PGE. Note that PGEs should *not* write to stdout or stderr.

- The **DpPrRusage** is the profiling program that outputs information about the runtime behavior of the PGE.
- Depending upon the PGE, it may take some time before the UNIX prompt returns.

5 At the UNIX prompt on the SDPS SGI, type **echo $status**, press **Enter**.

- The **$status** is an environment variable that stores the exit status of the previous program run, in this case, the PGE.
- A status of zero indicates success; a status of non-zero indicates an error of some kind.
- The meaning of a non-zero exit status should be documented and included with the DAPs.
- This command must be run ***immediately*** after the **DpPrRusage** command.

6 At the UNIX prompt on the SDPS SGI, type **vi *ResultsOut***, press **Enter**.

- The ***ResultsOut*** is the file name under which the profiling output was saved. Other output of the PGE may also be in this file.
- The **DpPrRusage** results may then be recorded and used when the PGE is registered in the PDPS.
- Any text editor/viewer may be used.

Examining PGE-Produced Log Files

Examining PGE-Produced Log Files

When the PGE is run outside of the PDPS, the PCF specifies the location and file names of the log files produced. This procedure describes how to locate that information from the PCF and use it to examine the log files.

The following is a list of tools, and or assumptions:

- The Data Processing Request ID (DPR ID) for PGE execution is required for this procedure.
- The SDP Toolkit writes the Log Status, Log User, and Log Report files to the directories specified in the PCF.
- All required environmental variables have been set.
- The operations/test account has read access to the PDPS Database.
- The .cshrc file should have the following entries.
    - setenv SYBASE/vendor/sybase
    - setenv SYROOT $SYBASE/sybooks
    - setenv EBTRC $SYBROOT/sun5m/.ebtrc
    - setenv DSQUERY computer-server (where computer-server is system dependent)
    - set path = ($path $SYBASE $SYBASE/bin $SYBROOT/sun5m/bin
- A text editor is available to the SSI&T personnel

LogStatus captures all error and status information concerning a program.

LogUser captures a subset of messages of level or type “_U_” or “_N_” which are of particular interest to a user.

LogReport captures arbitrary message strings sent by the PGE software. The messages are unrelated to the toolkit functions, but will contain important messages for scientists and developers.

Examining PGE-Produced Log Files for PGEs Run Outside of PDPS

---

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **cd *PCFpathname***, press **Enter**.

- The ***PCFpathname*** is the full path name to the location of the PCF used by the PGE for which log files are to be examined.

4 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *PCFfilename***, press **Enter**.

- The ***PCFfilename*** is the file name of the PCF used by the PGE for which log files are to be examined.
- This brings up the file named ***PCFfilename*** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs MOD35.pcf**, press **Enter**.

5 In the editor, search for logical IDs (beginning in the first column) **10100**, **10101**, and **10102**. These are the PCF entries for the Status log, User log, and Report log respectively. For each, note the file names in field 2 and the path names in field 3. Then quit the editor.

- If field 3 is blank, then the location is given by the default location specified in a line above the entries beginning with the "!" character.

6 At the UNIX prompt on the SDPS SGI, type **vi *StatusLogPathname/filename***, press **Enter**.

- The ***StatusLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10100. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs /PGE/MOD35/LogStatus**, press **Enter**.

7 At the UNIX prompt on the SDPS SGI, type **vi *UserLogPathname/filename***, press **Enter**.

- The ***UserLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10101. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs /PGE/MOD35/LogUser**, press **Enter**.

8 At the UNIX prompt on the SDPS SGI, type **vi *ReportLogPathname/filename***, press **Enter**.

- The ***ReportLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10102. When finished, quit the editor.
- Note any anomalous messages in file.
- Any text editor may be used such as *emacs*. For example, **/PGE/MOD35/LogReport**, press **Enter**.

Production History Log Files From PGEs Run Within the PDPS

The Production History (PH) is created during PGE execution within the PDPS and then Inserted into the IMF Data Server upon PGE completion. Included in the PH are the PGE log files. To access a Production History associated with a particular PGE run requires the DPR ID of the PGE run and the full path name to the IMF Data Server Production History archive.

---

1 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **cd *PHArchivePathname***, press **Enter**.

- The ***PHArchivePathname*** is the full path name of the IMF Data Server Production History archive.
- For example, type **cd /net/sprg1sgi/imf_data/archive/PH**, press **Enter**.

2 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **ls -al**, press **Enter**.

- A list of the current contents will be displayed. These will be Production History tar files. The names of these files consist of the DPR ID, the collection time, and the insertion time.
- Look for the PH corresponding to the DPR ID of interest.

3 At a UNIX prompt on the AIT Sun or on the SDPS SGI, type **cp *PHtarFilename WorkingPathname***, press **Enter**.

- The ***PHtarFilename*** is the file name of the Production History tar file.
- The ***WorkingPathname*** is the full path name to some working directory in which the Production History tar file is to be placed and examined.

4 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **cd *WorkingPathname***, press **Enter**.

- The ***WorkingPathname*** is the full path name to the working directory specified in step 3.

5 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **tar xvf *PHtarFilename***, press **Enter**.

- The ***PHtarFilename*** is the file name of the Production History tar file in the working directory.
- This command will untar the Production History tar file, extracting its component files into the current directory.

6 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *StatusLogFilename***, press **Enter**.

- The ***StatusLogFilename*** is the file name of the Status log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs LogStatus**, press **Enter**.

7 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *UserLogFilename***, press **Enter**.

- The ***UserLogFilename*** is the file name of the User log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs LogUser**, press **Enter**.

8 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *ReportLogFilename***, press **Enter**.

- The ***ReportLogFilename*** is the file name of the Report log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs LogReport**, press **Enter**.

This page intentionally left blank.

File Comparison

File Comparison Overview

The purpose of File Comparison is to ensure that the porting of the science software from the development facility at the SCF to the operational facility at the DAAC has not introduced any errors.

A number of file comparison tools are available during SSI&T via the SSIT Manager GUI or they can be invoked from the UNIX command line. Two tools are available for comparing HDF or HDF-EOS files, one tool for comparing ASCII files, and another tool for assisting in comparing binary files.

Using the GUI HDF File Comparison Tool

The procedures assume that two HDF or HDF-EOS files exist with similar structures, or at least having a data set in common. If either of the two HDF/HDF-EOS files is in a ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

To compare two HDF or HDF-EOS files, execute the procedure steps that follow:

1 Log onto the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter.**

2 Enter the **password** the press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter.**

The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Product Examination**, then **HDF**.

The HDF File Comparison Tool GUI will be displayed.

5 In the HDF File Comparison Tool GUI, click on the **File 1** button.

Read the *Systems Description* document and the *Operations Manual*. Both of these or their equivalent should be in the delivery.

Using the hdiff HDF File Comparison Tool

The procedures assume that two HDF or HDF-EOS files exist with similar structures, or at least having a data set in common. If either of the two HDF/HDF-EOS files is in a ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

To compare two HDF or HDF-EOS files, execute the procedure steps that follow:

1 Log onto the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter.**

2 Enter the **password** the press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter.**

- The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Product Examination**, then **File Comparison**, and then **HDF (hdiff)**.

- An xterm window running *hdiff* will be displayed.

5 In the xterm window at the prompt **Options? (-h for help)**, type in any desired options and then press **Enter**.

- To see the list of available options, type **-h** and press **Enter** to the prompt.

6 In xterm window at the prompt **1st file to compare?**, type ***filename1***, press **Enter.**

- The ***filename1*** is the file name of the first of two HDF or HDF-EOS files to be compared.
- If ***filename1*** is not in the current directory (the directory from which the SSIT Manage was run), include the full path name with the file name.

7 In xterm window at the prompt **2nd file to compare?**, type ***filename2***, press **Enter.**

- The ***filename2*** is the file name of the second of two HDF or HDF-EOS files to be compared.
- If ***filename2*** is not in the current directory (the directory from which the SSIT Manage was run), include the full path name with the file name. The two files will be compared and the output will be displayed in the xterm window.

Using the ASCII File Comparison Tool

Most output files (products) from PGEs run in the DAAC will be in HDF-EOS format. A small minority may be in ASCII (text) format. The ASCII File Comparison Tool is a front-end to *xdiff* UNIX X Window tool for comparing two ASCII files.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

The procedures assume that two ASCII files exist and have read permissions. If either of the two ASCII files are in a ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

To compare two ASCII files, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Product Examination**, then **File Comparison**, and then **ASCII**.

An xterm window running *xdiff* will be displayed.

2 In xterm window at the prompt **1st file to compare?**, type ***filename1***, press **Enter.**

The ***filename1*** is the file name of the first of two ASCII files to be compared.

If ***filename1*** is not in the current directory (the directory from which the SSIT Manage was run), include the full path name with the file name.

3 In xterm window at the prompt **2nd file to compare?**, type ***filename2***, press **Enter.**

The ***filename2*** is the file name of the second of two ASCII files to be compared.

If ***filename2*** is not in the current directory (the directory from which the SSIT Manager was run), include the full path name with the file name.

A GUI labeled **xdiff** will be displayed.

4 In the GUI labeled **xdiff**, view the differences between the two files displayed.

File ***filename1*** will be displayed on the left side of the GUI. File ***filename2*** will be displayed on the right.

Only sections of file in which there are differences will be displayed. A "bang" character (!) at the beginning of a line indicates that a difference was found.

For further help on *xdiff*, type **man xdiff**, press **Enter** in an xterm window.

Close the display window by using the pull down menu from the X window in the upper left corner.

5 In the xterm window at the prompt **Hit Enter for another diff, 'q <Enter>' to quit:**, type **q** and then press **Enter** to quit or just press **Enter** to perform another comparison.

Using the Binary File Difference Assistant

Most output files (products) from PGEs run in the DAAC will be in HDF-EOS format. A small minority may be in some binary format. The Binary File Difference Assistant aids the user in

constructing code that allows comparison of binary output files. Since there is an unwieldy number of possibilities for binary file formats, this tool cannot compare two binary files without some custom code written at the DAAC, hence, the "Assistant" in the name. The Binary File Difference Assistant aids the user by generating a makefile, a driver module, and a template comparison module in C, FORTRAN 77 or IDL (Interactive Data Language). The user then edits these templates to read the particular binary format in question according to a SCF-supplied format specification.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

The procedures assume that two files exist and that information on the data structure and tolerances have been provided.

To compare two binary files, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Product Examination**, then **File Comparison**, and then **Binary**.

- The Binary File Difference Assistant tool GUI will be displayed.

2 In the Binary File Difference Assistant tool GUI, click on one of the languages listed under the **Select Language** label. The choices are C, FORTRAN, or IDL.

- The choice of language depends largely on preference. It does not necessarily have to be the language that was used to create the files being compared.

3 Optionally, click on either the **Image** button or the **Structure** button located under the label **Compare Function**.

- Clicking on the **Image** button will display a code example for comparing binary files containing images.
- Clicking on the **Structure** button will display a code example for comparing binary files containing structures or records.
- The displayed listing well documented and should be read.
- The language of the code will depend on the language selection made in step 2.

4 Optionally, click on either the **Image** button or the **Structure** button located under the label **Driver**.

- Clicking on the **Image** button will display a code example for a driver invoking the compare function for binary files containing images.
- Clicking on the **Structure** button will display a code example for a driver invoking the compare function for binary files containing structures or records.

- The displayed listing well documented and should be read.
- The language of the code will depend on the language selection made in step 2.

5 Optionally, click on either the **Help** button.

- A Help GUI will be displayed.
- To end help, click on the **Dismiss** button.
- The Help GUI may remain displayed while using the Binary File Difference Assistant.

6 Once familiar with the code examples (steps 3 and 4), click on the **Copy** button.

- A GUI labeled **Enter Unique ID** will be displayed.
- In the field labeled **Enter unique file identifier:**, type ***fileID***, click on the **OK** button.
- The ***fileID*** will be used in the file names of the files copied over. These files will be:

  C:

  - DaacBinDiff_***fileID***.c Compare function
  - DaacBinDiff_***fileID***_driver.c Driver
  - DaacBinDiff_***fileID***.mak Makefile

  FORTRAN:

  - DaacBinDiff_***fileID***.f Compare function
  - DaacBinDiff_***fileID***_driver.f Driver
  - DaacBinDiff_***fileID***.mak Makefile

  IDL:

  - DaacBinDiff_***fileID***.pro Compare function
  - DaacBinDiff_***fileID***_driver.pro Driver
  - DaacBinDiff_***fileID***.sh Shell script with here document

- The files will be copied into the directory from which the SSIT Manager is being run.

7 Using any desired text editor, customize the files for the job at hand. Then build the executable using the customized makefile provided (for C and FORTRAN). Then run the program to perform the binary file comparison.

This page intentionally left blank.

Data Visualization

Another feature of the SSIT Manager is the EOS view. EOSView was developed by ECS for creating two-dimensional displays from HDF-EOS objects (Grid, Swath) as well as the standard HDF objects (SDS, Vdata, Image, Text). It has additional features such as thumbnail-panning, colorization, zooming, plotting, and animation. It is a user-friendly visualization tool for a user to inspect data sets produced by PGEs. Only some aspects of data visualization will be addressed in this training material. For further information, see the related references.

Viewing Product Metadata with the EOSView Tool

This procedure describes how to use the EOSView tool to inspect the metadata in the HDF-EOS output file from a PGE. To view product metadata with the EOSView tool, execute the procedure steps that follow:

---

1 From the SSIT Manager, click on the **Tools** menu, then choose **Product Examination**, then **EOSView**. The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Open.** The **Filter** GUI will be displayed.

- In the subwindow labeled **Filter**, select the appropriate directory and file to open.

3 A GUI labeled **EOSView - *MyOutputFile.hdf*** will be displayed where *MyOutputFile.hdf* is the file name of the file chosen in step 2. Once displayed, a list of HDF objects will appear in the main window. If nothing is listed, it means that no HDF objects were found within the file.

4 In the GUI labeled **EOSView - *MyOutputFile.hdf*,** click on an object listed for which metadata is to be inspected. The object selected will be highlighted.

- Do not double click on object since this will cause a **Dimension** GUI to be displayed instead.

5 The global metadata associated with the object selected will be displayed in a scrollable field by clicking on the **Attributes** menu and selecting **Global** in the GUI labeled **EOSView - *MyOutputFile.hdf*.**

- If instead, the message “Contains no Global Attributes” appears, then the selected object contains no global metadata.

6 Repeat steps 4 and 5 for each HDF object within the selected HDF-EOS file for which metadata is to be examined.

7 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **File** menu and select **Close** to close the **EOSView - *MyOutputFile.hdf*** GUI.

8 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Exit** to exit **EOSView - EOSView Main Window** GUI.

Viewing Product Data with the EOSView Tool

This procedure describes how to use the EOSView tool to inspect the science data in the HDF-EOS output file from a PGE. To view product data with the EOSView tool, execute the procedure steps that follow:

---

1 From the SSIT Manager, click on the **Tools** menu, then choose **Product Examination**, then **EOSView**. The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Open.** The **Filter** GUI will be displayed.

- In the subwindow labeled **Filter**, select the appropriate directory and file to open.

3 A GUI labeled **EOSView - *MyOutputFile.hdf*** will be displayed where *MyOutputFile.hdf* is the file name of the file chosen in step 2. Once displayed, a list of HDF objects will appear in the main window.

- If nothing is listed, it means that no HDF objects were found within the file.

4 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on an object listed for which data is to be inspected.

5 Go to procedure depending upon the type of the object selected in step 4.

- Proceed to Section 1.2.1 if the selected object is an HDF Image.
- Proceed to Section 1.2.2 if the selected object is an HDF-EOS Grid.
- Proceed to Section 1.2.3 if the selected object is an HDF-EOS Swath.
- Proceed to Section 1.2.4 if the selected object is an HDF SDS.

6 Repeat steps 4 and 5 for each HDF object within the selected HDF-EOS file for which data is to be examined.

7 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **File** menu and select **Close** to close the **EOSView - *MyOutputFile.hdf*** GUI.

8 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Exit** to exit the **EOSView - EOSView Main Window** GUI.

Viewing HDF Image Objects

This procedure describes how to use the EOSView tool to view science Images in the HDF-EOS output file from a PGE. The procedures assume that the output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools and that at least one object is an HDF image (RIS8, RIS24, *i.e.* Browse data). It is also assumed that EOSView has been properly installed and is accessible to the user and that the HDF-EOS file has been read into EOSView.

To view an HDF-EOS Image object with the EOSView tool, execute the procedure steps that follow:

1 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on an Image object listed for which data is to be inspected.

- A GUI labeled **EOSView - Image Display Window - *MyImageObject*** will be displayed where *MyImageObject* is the name of the object selected.

2 Optional colorization. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **Palette** menu, then select **Select** and then select one of the palettes listed: **Default**, **Greyscale**, **Antarctica**, **Rainbow**, or **World Colors**..

- This selection may be repeated until the desired palette is chosen.

3 Optional zooming. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **Zooming** menu, then select **Select** and then select one of the resampling methods listed: **Bilinear Interpolation** or **Nearest Neighbor**. Then click on the **Zoom In** or **Zoom Out** buttons to apply the method.

- The zooming options may be repeated as desired.

4 Optional panning while zooming. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **Options** menu, then select **Pan Window**, a thumbnail representation of the entire Image will be displayed in the subwindow labeled **Pan Window**. The portion of the zoomed Image shown in the main window will be the portion indicated by the hollow rectangle on the thumbnail image. Use the mouse left button to click and drag the rectangle to a new location on the thumbnail image.

- The panning option may be repeated as desired.

5 To end the session with colorization, zooming, or panning, in the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **File** menu and select **Close**.

6 Optional animation. In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **Options** menu, then select **Animated images**.

- A GUI labeled **EOSView - Image Animation Window - *MyOutputFile.hdf*** will be displayed.
- Optionally, click on the **Options** menu and then select **Mode** to select how the animation is to be run. Choose **Stop at end**, **Continuous run**, or **Bounce**.
- To end animation session, click on the **File** menu and then select **Close**.

Viewing HDF-EOS Grid Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF-EOS Grid format. These are generally the science data and not browse images.

To view an HDF-EOS Grid object with the EOSView tool, execute the procedure steps that follow:

---

1 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on an Grid object listed for which data is to be inspected. A GUI labeled **EOSView - Grid Select** will be displayed.

- Information on **Grid Information, Projection Information, Dimensions, Attributes** for the selected object can be displayed by clicking on the appropriate checkboxes.

2 In the GUI labeled **EOSView - Grid Select**, click on the **Data Fields** checkbox and then click on the **OK** button. Then double click on one of the data fields listed.

- A GUI labeled **EOSView - Grid - *GridObjectName* - Start/Stride/Edge** will be displayed where *GridObjectName* will be replaced by the name of the Grid object selected in step 1.

3 To display the data in the form of a table of values, in the GUI labeled **EOSView - Grid - *GridObjectName* - Start/Stride/Edge**, click on the checkboxes for both **YDim** and **XDim** and then click on the **OK** button.

- A GUI labeled ***MyDataField*** will be displayed where *MyDataField* will be replaced by the name of the data field selected in step 2.

4 To display the data field in image form, in the GUI labeled ***MyDataField***, click on the **File** menu and then select **Make Image**. A GUI labeled **EOSView - Swath/Grid Image** will appear,

5 Optional colorization, zooming, panning while zooming can be used to obtain your desired output.

6 To end the session with displaying Grid object, in the GUI labeled **EOSView - Swath/Grid**, click on the **File** menu and select **Close**. The **EOSView - Swath/Grid** GUI will disappear.

Viewing HDF-EOS Swath Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF-EOS Swath format. These are generally the science data and not browse images.

To view an HDF-EOS Swath object with the EOSView tool, execute the procedure steps that follow:

---

1 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on a Swath object listed for which data is to be inspected. A GUI labeled **EOSView - Swath Select** will be displayed.

- Information on **Dimensions, Geolocation Mappings, Indexed Mappings, Geolocation Fields**, **Attributes** for the selected Swath Object can be displayed by clicking on the corresponding checkboxes.

2 In the GUI labeled **EOSView - Swath Select**, click on the **Data Fields** checkbox and then click on the **OK** button. Then double click on one of the data fields listed.

- A GUI labeled **EOSView - Swath - *SwathObjectName* - Start/Stride/Edge** will be displayed where *SwathObjectName* will be replaced by the name of the Swath object selected in step 1.

3 To display the data in the form of a table of values, in the GUI labeled **EOSView - Swath - *SwathObjectName* - Start/Stride/Edge**, click on the checkboxes for both **ScanLineTra** and **PixelsXtrac** and then click on the **OK** button.

4 To display the data field in image form, in the GUI labeled ***MyDataField***, click on the **File** menu and then select **Make Image**. A GUI labeled **EOSView - Swath/Grid Image** will appear,

5 Optional colorization, zooming, panning while zooming features can be used in the GUI labeled **EOSView - Swath/Grid Image** to obtain your desired image.

6 To end the session with displaying Swath object, in the GUI labeled **EOSView - Swath/Grid**, click on the **File** menu and select **Close**.

Viewing HDF SDS Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF SDS (standard HDF science data set) format. To view an HDF SDS object with the EOSView tool, execute the procedure steps that follow:

---

1 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on a SDS object listed for which data is to be inspected. A GUI labeled **EOSView - Multi-Dimension SDS** will be displayed.

- A number of checkboxes will be displayed, one for each of the dimensions in the selected SDS (there will be at least two, an X and a Y).

2 In the GUI labeled **EOSView - Multi-Dimension SDS**, click on two of the dimension checkboxes and then click on the **Table** button. Then double click on one of the data fields listed.

- A GUI labeled ***MySDS*** will be displayed where *MySDS* will be replaced by the name of the SDS object selected in step 1.

3 To display the data field in image form, in the GUI labeled ***MySDS***, click on the **File** menu and then select **Make Image**. A GUI labeled **EOSView - Image Display Window - *MySDS*** will appear,

4 Optional colorization, zooming, panning while zooming can be used to obtain your desired output.

5 To end the session with displaying Swath object, in the GUI labeled **EOSView - Image Display Window - *MySDS***, click on the **File** menu and select **Close**.

- The **EOSView - Image Display Window - *MySDS*** GUI will disappear.

Updating the PDPS Database and IMF Data Server

PDPS Database Update

Once the PGE has been compiled and the build completed and CCB approved, the Production Planner is notified and the use of the PGE is authorized and placed on the Database and Data Server.

The PDPS Database Update Tools are accessed from the TOOLS menu item of the SSIT Manager GUI tool bar, submenu PDPS Database. This tool is used to register a PGE with the PDPS database, during which critical information about the PGE and under what conditions it should be run is captured. This "PDPS metadata" is used by the PDPS Processing and Planning subsystems prior to and during PGE execution, both in the test environment and in production (after the data is copied to the production machine.)

All programs may be started from the TOOLS:PDPS Database menu.

PCF ODL Template

The first step in PDPS Database Update is to generate a template PDPS/SSIT PGE SCIENCE metadata ODL file from the PCF delivered with the science software. The user is prompted for input, including the name of a delivered science software Process Control File (PCF), PGE name and version.

After the template file is generated, the SSIT operator must edit this file, adding all information which is currently blank, as needed. In particular, each PCF file entry must have a "Data Type Name", a.k.a. ESDT name, a.k.a. Data Server Short name.

Creating a PCF ODL Template File

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 Select the **Tools→PDPS Database→ PCF ODL Template** from the SSIT Manager menu bar.

5 Enter the name of the delivered science software PCF by typing: **PGE_<PGF name>#<PGE version>.odl** where **PGE name** is the name of the PCF file and PGE version is its version.

6 Save the template file by selecting the **File→Save** from the menu bar.

7 Update the template file by adding all information which is currently blank.

8 Close the template file by selecting the **File→Close** from the menu bar.

- You will be prompted so **Save** the file, select the **Yes** button.

Update PDPS/SSIT Database with PGE And ESDT Science Metadata

The next step is to update the PDPS/SSIT database with SCIENCE metadata, where the latter is defined as PDPS metadata that comes from the Instrument Teams (ITs), and rarely changes.

The PGE template ODL file, which is output from the *PCF ODL Template* program of the previous section, is input to this program, after it has been edited and renamed.

The procedures assume that the environment variable DPAT ESDT SCIENCE METADATA points to a directory used for containing the PDPS ESDT metadata ODL files.

Updating PDPS/SSIT Database with ESDT Metadata

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Determine the IMF Data Server ShortName for the ESDT corresponding to the given file. One place to look for ESDT names is the Earth Sciences Online Directory.

4 At a UNIX prompt on the AIT Sun, type **ls $DPAT_ESDT_SCIENCE_MD/ESDT_*ShortName*.odl**, press **Enter**.

- The **$DPAT_ESDT_SCIENCE_MD** is an environment variable containing the full path name to the location of existing ESDT ODL files.
- The **ESDT_*ShortName*.odl** is the file name of the ESDT ODL file you are looking for where ***ShortName*** is the ESDT's ShortName. If a file for the desired ESDT is listed, then it has already been prepared and this procedure can be exited now.
- For example, if the desired ESDT has the ShortName ca1182, type **ls $DPAT_ESDT_SCIENCE_MD/ESDT_ca1182.odl,** press **Enter**.
- If the desired file is *not* listed, continue on to step 3.

5 At a UNIX prompt on the AIT Sun, type **cd *WorkingPathname***, press **Enter.**

- The ***WorkingPathname*** is the full path name to a working directory for which the user has write permissions.
- For example, **cd /home/jdoe/working/**, press **Enter**.

6 At a UNIX prompt on the AIT Sun, type **cp /usr/ecs/Rel_A/CUSTOM/data/DPS/ESDT SHRTNAME.odl.tpl ESDT_ShortName.odl**, press **Enter**.

- The **ESDT_*ShortName*.odl** is the file name of the ESDT ODL file to be created.
- This command copies a template ESDT ODL file to the ESDT ODL file to be created. The template is well commented.
- For example, type **cp /usr/ecs/Rel_A/CUSTOM/data/DPS/ESDT_SHRTNAME.odl.tpl ESDT_ca1182.odl,** press **Enter**.
- The **ESDT_*ShortName*.odl** file naming convention *must* be observed.

7 At a UNIX prompt on the AIT Sun, type **vi ESDT_*ShortName*.odl**, press **Enter**.

- The **ESDT_*ShortName*.odl** represents the file name of the ESDT ODL template file created in step 4.
- Any text editor may be used such as *emacs*. For example, **emacs ESDT_ca1182.odl**, press **Enter**.

8 In the file, add required metadata to the ODL template.

- Use the internal documentation contained in the ODL file (from the original template) to aid in populating with metadata.
- Note that the ShortName specified within the file must match the ShortName of the file name itself.
- In addition, the ShortNames used in the PDPS PGE metadata ODL file must match the ShortNames in these files.

9 Save the changes made to the ESDT metadata ODL file and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Enter**.
- For other editors, refer to that editor's documentation.

10 At a UNIX prompt on the AIT Sun, type **cp ESDT_*ShortName*.odl $DPAT_ESDT_SCIENCE_MD**, press **Enter**.

- For example, in the case where the ESDTs ShortName is ca1182, type **cp ESDT_ca1182.odl $DPAT_ESDT_SCIENCE_MD**, press **Enter**.
- This copies the newly constructed ESDT metadata ODL file to the proper location.

11 Repeat steps 1 through 8 for each ESDT required by a particular PGE. When all ESDT metadata ODL files have been completed, continue on to Section 11.1.2.

Updating PDPS/SSIT Database with PGE Metadata

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start the SSIT Manager GUI by typing SSIT script alias **[**alias name**]** then press **Enter**.

- The SSIT Manager GUI is displayed.

4 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **PCF ODL Template**.

- An xterm in which DpAtCreateOdlTemplate.sh is running will be displayed.
- Alternatively, the same tool can be invoked by typing at a UNIX prompt on the AIT Sun **DpAtCreateOdlTemplate.sh**, press **Enter**.

5 At the program prompt **Config filename (default *defaultConfigFile*)?**, press **Enter**.

- The default configuration file for this tool will be used.
- The ***defaultConfigFile*** will be replaced by the full path name and file name of the default configuration file. The file name will be DpAtCS_*daac*.CFG where *daac* will be replaced by one of {GSFC, EDC, LARC, NSIDC}.

6 At the program prompt **Process Control file name?**, type ***PCFpathname*/*PCFfilename***, press **Enter**.

- The ***PCFpathname*** is the full path name to the location of the PCF. If not specified, the directory from which the SSIT Manager was run will be assumed.
- The ***PCFfilename*** is the file name of the PCF.

7 At the program prompt **PGE name?**, type ***PGEname***, press **Enter**.

- The ***PGEname*** is the name of the PGE that will be registered.

8 At the program prompt **PGE version (default 1)?**, type ***PGEversion***, press **Enter** or just press **Enter** if the default shown is correct.

- The ***PGEversion*** is the version of the PGE that will be registered.
- After a brief time, the message "Successfully created ODL template file" should be displayed if the task was successful.
- The program will output a file with the filename **PGE*_PGEname#PGEversion*.tpl**.
- For example, if the PGE name was **MOD35**, and the version was **1**, this output file will be named **PGE_MOD35#1.tpl**.

9 At the program prompt **Hit Enter to run again, 'q <Enter>' to quit:**, press **Enter** to repeat process with another PCF or type **q** and press **Enter** to quit.

- The xterm will disappear.

10 At a UNIX prompt on the AIT Sun, type **cd *SSITrunPathname***, press **Enter.**

- The ***SSITrunPathname*** is the full path to the directory from which the SSIT Manager was run. This will be the directory where the file **PGE_*PGEname#PGEversion*.tpl** will reside.

11 At a UNIX prompt on the AIT Sun, type **cp PGE_*PGEname#PGEversion*.tpl PGE_*PGEname#PGEversion*.odl**, press **Enter.**

- The **PGE_*PGEname#PGEversion*.tpl** is the file name of the ODL template file created in step 8.
- The **PGE_*PGEname#PGEversion*.odl** is the file name of a copy which can be safely edited. This file name convention must be used.

12 At a UNIX prompt on the AIT Sun, type **vi PGE_*PGEname#PGEversion*.odl**, press **Enter.**

- The **PGE_*PGEname#PGEversion*.odl** is the file name of the copy created in step 11.
- Any text editor may be used such as *emacs*. For example, **emacs MOD35#1.odl**, press **Enter**.

13 In the file, add required metadata to the ODL template.

- For an explanation of what metadata is required, see file /usr/ecs/Rel_A/CUSTOM/data/DPS/PGE_ssit#11.tpl.
- Note that the ShortNames typed into this file must each have a corresponding PDPS ESDT metadata ODL file.

14 Save the changes made to the ODL template file and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Enter**.
- For other editors, refer to that editor's documentation.

15 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **SSIT Science Metadata Update**.

- An xterm in which DpAtPdpsDbUpdateScience.sh is running will be displayed.
- Alternatively, the same tool can be invoked by typing at a UNIX prompt on the AIT Sun **DpAtPdpsDbUpdateScience.sh**, press **Enter**.

16 At the program prompt **Config filename (default *defaultConfigFile*)?**, press **Enter**.

- The default configuration file for this tool will be used.
- The ***defaultConfigFile*** will be replaced by the full path name and file name of the default configuration file. The file name will be DpAtDS_*daac*.CFG where *daac* will be replaced by one of {GSFC, EDC, LARC, NSIDC}.

17 At the program prompt **mode (default *defaultMode*)?**, type ***mode***, press **Enter** or just press **Enter** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **ssit**.

18 At the program prompt **PGE name?**, type ***PGEname***, press **Enter**.

- The ***PGEname*** is the name of the PGE that will be registered. This name must match the PGE name specified in step 4.

19 At the program prompt **PGE version (default 1)?**, type ***PGEversion***, press **Enter** or just press **Enter** if the default shown is correct.

- The ***PGEversion*** is the version of the PGE that will be registered. This version must match the PGE version specified in step 5.
- The PDPS database will then be updated with the information contained in the file **PGE_*PGEname#PGEversion*.odl**

20 To the final prompt **Hit Enter to run again, 'q <Enter> to quit:**, press **Enter** to update the PDPS database with another PGE ODL metadata file or type **q** and press **Enter** to quit.

- If you make a mistake entering any values, press **Enter** here; your previous entries are restored as defaults and you won't have to retype them.

SSIT Operational Metadata Update GUI

The SSIT Operational Metadata Update GUI (Figure 6) is used to view or update the following operational parameters for a particular PGE:

- Performance parameters for the PGEs.
- Resource parameters for the PGEs.
- PGE user-defined static parameter.
- View the PGE science metadata file.

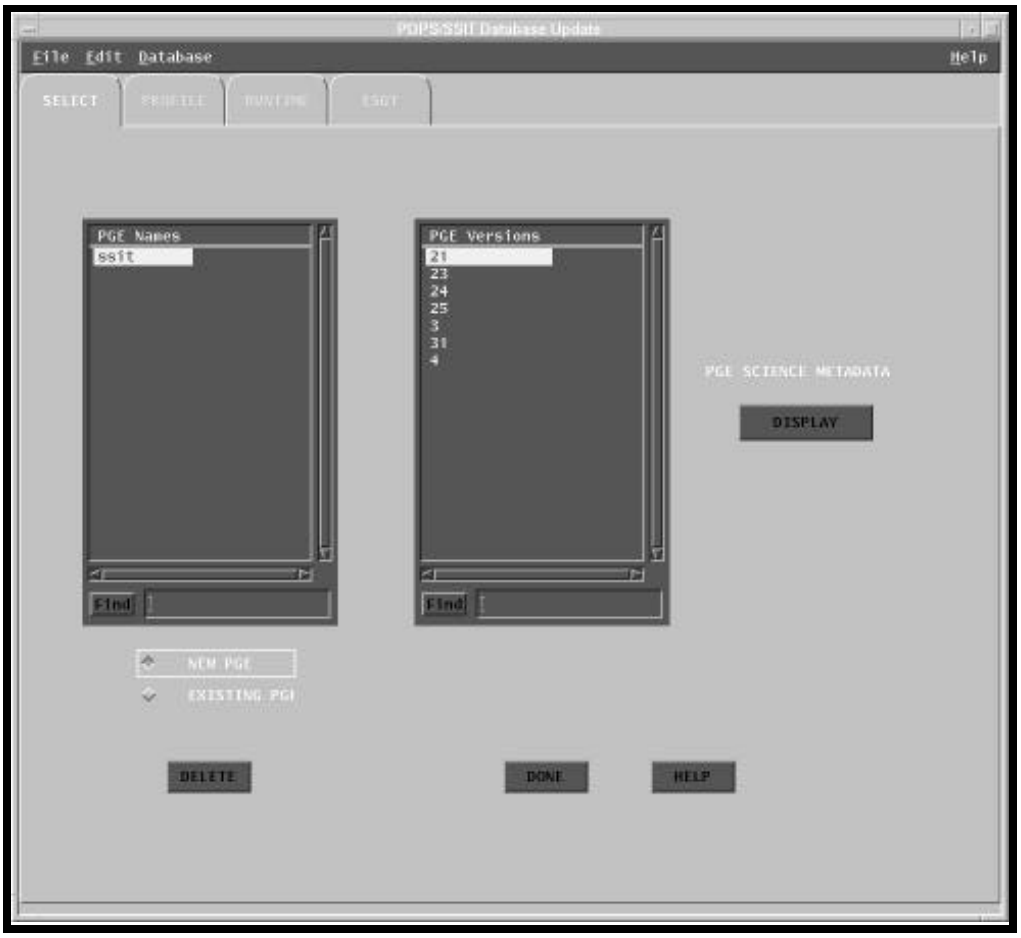

Figure 6. SSIT Database Operational Metadata Update GUI – SELECT view

The SSIT Database Operational Update GUI has four different options to select. Each tab represents different views that can be selected which permit the operator to display/modify various fields/parameters contained in a given PGE. The following provides a description for the tabs available on this GUI:

- **SELECT view -** The Select tab provides a method to selecting either New or Existing PGEs. Selecting a PGE from the list populates the version list with versions of the selected PGE. Once a version is selected, user gains access to the remaining tabs (Profile, Runtime, and ESDT).
- **PROFILE view -** Selecting the Profile tab displays the profile information of the selected PGE. The view provides access to the Resource and Performance fields for the selected PGE. These fields may be edited or added but must be populated with data prior to the start of the database update.
    - Wall clock Time - real time in seconds
    - CPU Time - CPU time used by the process

- Max Memory Used - maximum memory used in megabytes
- Block Input Ops - number of times the file system performed input servicing
- Block Output Ops - number of times the file system performed output servicing
- Swaps - number of times the process was swapped out of main memory
- Page Faults - number of page faults serviced by the process
- Disk Space used in PGE run - maximum disk space used by the process
- Processing String - a name given to collection of processors where PGE can be executed
- Computer Name - name of the computer where the process is executed
- Number of Processors - number of processors

- **RUNTIME view -** Selecting the Runtime tab displays the static parameters used by the PGE at runtime. The left screen represents data that has already been entered in the database. On the right side of the screen, the user may enter descriptive text for each runtime parameter. Runtime parameter descriptions are optional.
- **ESDT view -** Selecting the ESDT tab displays the ESDT information of the selected PGE. The ESDTs for the selected PGE are displayed in a list. Once selected, the associated PDPS Science ESDT metadata can be viewed.

Updating SSIT Metadata

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Change directories to the location of the SSIT Manager GUI, by typing the following UNIX command line in the XTerm window.

cd /usr/ecs/Rel_A/CUSTOM/bin/DPS then press the **Enter** key.

- The changed directory path is displayed in front of a new command line.

4 Invoke the SSIT Manager GUI by typing the following UNIX command line in the XTerm window.

DpAtMgr ConfigFile /usr/ecs/Rel_A/cfg/DpAtMG_GSFC.CFG & then press the **Enter** key.

- The SSIT Manager GUI is displayed.

5 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **SSIT Opnl Metadata Update**.

- The PDPS/SSIT Database Update GUI will be displayed.

6 In the subwindow labeled **PGE Names**, click on a PGE name. Then in the subwindow labeled **PGE Versions**, click on the PGE version for that PGE.

- The PGE name and version will be highlighted when you click on them.
- If the PGE name and/or version does not appear in the lists, it means that the updating of the PDPS database with PGE metadata was not successful.

7 Click on the radio button labeled **NEW PGE** in the lower left quadrant. Then click on the button labeled **DONE**.

- The page tabs **PROFILE**, **RUNTIME**, and **ESDT** will change from gray (indicating disabled) to black (indicating enabled).

8 Click on the **PROFILE** page tab.

- The Profile page will be displayed.

9 In the fields under the label **Performance Statistics**, enter the information specified.

- In the field labeled Wall clock time, enter the amount of wall clock time it takes for one execution of the PGE, in seconds. The tab **PROFILE** will change from black (indicating enabled) to red (indicating database needs to be updated by APPLY button).
- In the field labeled **CPU time (user)**, enter the so-called *user* time of the PGE, in seconds. This value should come from profiling the PGE (see Section 10.2).

- In the field labeled **Max memory used**, enter the maximum amount of memory used by the PGE, in megabytes (MB). This value should come from profiling the PGE (see Section 10.2).

- In the fields labeled **Block input ops** and **Block output ops**, enter the integer number of block inputs and block outputs, respectively. These values should come from profiling the PGE (see Section 10.2).

- In the field labeled **Swaps**, enter the integer number of page swaps from the PGE. This value should come from profiling the PGE (see Section 10.2).

- In the field labeled **Page faults**, enter the integer number of page faults from the PGE. This value should come from profiling the PGE (see Section 10.2).

10 In the fields under the label **Resource Requirements**, enter the information specified.

- In the field labeled **Max. DISK SPACE used during PGE run**, enter the maximum amount of disk used by the PGE during execution, in megabytes (MB). Space should be allowed for the executable(s), input files, output files, ancillary files, static files, MCFs, and the PCF. (This number should also be in the PGE metadata ODL file; yes, there is duplication here.)

- Click on the radio button labeled **Proc. String** (if not already clicked on).

- A list of processing strings should appear in the scrollable window to the left of the two radio buttons **Proc. String** and **Computer Name**. Nominally, only one item should be listed and should be highlighted.

- In the field labeled **No Processors**, the number 1 should appear. In the Testbed, this number is always 1, and cannot be changed. This is a placeholder for Release B.0.

11 Once the fields on the **PROFILE** page have been completed, click on the **APPLY** button.

- This will update the PDPS database with the information just entered. The tab **PROFILE** will change from red (indicating database needs to be updated) to black (indicating enabled).

- To start over, click on the **RESET** button. This will clear all fields.

12 Click on the **File** menu and select **Exit**.

- This will end the session with PDPS/SSIT Database Update and the GUI will disappear.

Copying PDPS Metadata to the PDP/Production Database

The final step in PDPS database update is to copy a single PGE version's worth of PDPS metadata from the SSIT database to the Production database. This is done after all SSI&T activities for this PGE version are finished, and the PGE is ready for promotion to Production status.

Copying a single PGE's metadata from SSIT Database to the Production Database

1	Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.
2	Enter the **password** then press **Enter.**
3	Change directories to the location of the SSIT Manager GUI, by typing the following UNIX command line in the XTerm window. **cd /usr/ecs/Rel_A/CUSTOM/bin/DPS** then press the **Enter** key. • The changed directory path is displayed in front of a new command line.
4	Invoke the SSIT Manager GUI by typing the following UNIX command line in the XTerm window. **DpAtMgr ConfigFile /usr/ecs/Rel_A/cfg/DpAtMG_GSFC.CFG &** then press the **Enter** key. • The SSIT Manager GUI is displayed.
5	Select the **Tools→Xterm** from the SSIT Manager menu bar. • An Xterm window is displayed.
6	Access the Copy Unix script by typing: **DpAtPromotePgeDbEntry.sh** and press the **Enter** key. • The **Username** and **Password** prompts are displayed .
7	Enter the Username and Password and press the **Enter** key. • The **PGE name** prompt is displayed.
8	Enter the **PGE name** and press the **Enter** key. • The **PGE version** prompt is displayed.
9	Enter the **PGE version** and press the **Enter** key. • An ISQL script is initiated which copies the PGE version from SSIT to the Production database. • Note that data for a particular PGE version can only be copied once. This is reasonable, because any time a PGE has been changed, it must have a new version.

Inserting Static Files to the Data Server

This program is used to insert PGE input files to the Data Server. These files are "static", i.e., they rarely (if ever) change for successive PGE runs. An example of such a file is a calibration coefficients file. Such files are normally part of the DAP.

Before this program is used, the following must have occurred:

- An ESDT is defined for the file.
- A file containing ASCII metadata for this ESDT granule must be constructed, in the format that the Data Server expects, i.e., a "target MCF".

Creating a Metadata ODL File for a Static Granule

1 Log into the AIT 1 Sun platform by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun, type **cd *WorkingPathname***, press **Enter**.

- The ***WorkingPathname*** is the full path name of the working directory containing the template metadata ODL file. For example, **cd /Rel_A/data/TemplateODL/**, press **Enter**.

4 At the UNIX prompt on the AIT Sun, type **cp *StaticODLmet.tpl filename*.met**, press **Enter**.

- The ***StaticODLmet.tpl*** is the file name of the template Target MCF.
- The ***filename*.met** is the file name of the Target MCF for this static file. The file name extension must be .met.
- This command will copy the template Target MCF to ***filename*.met**. For example, type **cp StaticODLmet.tpl CER11T.mcf.met**, press **Enter**.

5 At a UNIX prompt on the AIT Sun, type **vi *filename*.met**, press **Enter.**

- This command invokes the *vi* editor and reads in the Target MCF created in step 2.
- Any text editor may be used such as *emacs*. For example, **emacs CER11T.met**, press **Enter**.

6 Edit the Target MCF with the specific information for the static data granule to be Inserted. The following guidelines should be followed when editing on the template MCF:

- The value for the SHORTNAME object should be filled out with proper instrument name. For example: " CER_MISC".

- The value for the Version ID object should be filled out with the proper version number. For example: “1” .
- In the INFORMATIONCONTENTCONTAINER object,
  - The value for the PARAMETERNAME object of the class “1” should be filled out with the name of static data file. For example: “CER11T.mcf”.
  - The value for the PARAMETERVALUE object of the class “2” should be filled out based on the following guideline:
    - If the data granule is a coefficient file, a “C” followed by a numerical number n (n=1,2,….) will be used. Here n stands for the number of the coefficient file. For example, if the first coefficient file is involved, “C1” will be used.
    - If the data granule is a MCF file, a “M” followed by a numerical number n (n=1,2,….) will be used. Here n stands for the number of the MCF file. For example, if the first MCF file is involved, “M1” will be used.
- A template Target MCF and a sample Target MCF are shown in Listings 11.3.1-1 and 11.3.1-2, respectively.

7 Save the changes made to the Target MCF (***filename*.met**) and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Enter**.
- For other editors, refer to that editor’s documentation.

Inserting Static Data Granules into the IMF Data Server

---

1 Log into the AIT 1 Sun platform by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Change directories to the location of the SSIT Manager GUI, by typing the following UNIX command line in the XTerm window.

cd /usr/ecs/Rel_A/CUSTOM/bin/DPS then press the **Enter** key.

- The changed directory path is displayed in front of a new command line.

4 Invoke the SSIT Manager GUI by typing the following UNIX command line in the XTerm window.

DpAtMgr ConfigFile /usr/ecs/Rel_A/cfg/DpAtMG_GSFC.CFG & then press the **Enter** key.

- The SSIT Manager GUI is displayed.

5 At the UNIX prompt on the AIT Sun, type **cd *DataGranulePathname***, press **Enter**.

- The ***DataGranulePathname*** is the full path name of the directory containing the static data granule(s) to Insert to the IMF Data Server.. For example, **cd /disk2/PGE32/data/**, press **Enter**.

6 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert Static**.

- An xterm in which DpAtInsertStaticFile.sh is running will be displayed.
- Alternatively, the same tool can be invoked by typing at a UNIX prompt on the AIT Sun **DpAtInsertStaticFile.sh**, press **Enter**.
- The DpAtInsertStaticFile.sh program will prompt for further information.

7 At the program prompt **Config filename (default *defaultConfigFile*)?**, press **Enter**.

- The default configuration file for this tool will be used.
- The ***defaultConfigFile*** will be replaced by the full path name and file name of the default configuration file. The file name will be DpAtIS_*daac*.CFG where *daac* will be replaced by one of {GSFC, EDC, LARC, NSIDC}.

8 At the program prompt **mode (default ops)?**, press **Enter**

- If **ops** is not the correct mode, then enter in the correct mode (*e.g.* ssit) and press **Enter**.

9 At the program prompt **ESDT name?** type ***ESDTShortName***, press **Enter**

- The ***ESDTShortName*** is the ShortName of the ESDT descriptor file corresponding to this granule to be Inserted. For example, type **CADMERLW**, press **Enter**.

10 At the program prompt **Science group?**, type ***ScienceGroupID***, press **Enter**

- The ***ScienceGroupID*** is an identifier used to define the file type as a coefficient file, a lookup table file, or a MCF. It distinguishes static granules of different types which share the same ESDT. For instance, for a coefficient file, use **C*n***, where number *n* could be 0, 1, 2…; this number *n* needs to be matched with the number *n* in the PGE_PGEName#Version.odl file (see Section 11.1.2). For example, type **C1**, press **Enter**.

11 At the program prompt **PGE name?**, type ***PGEName***, press **Enter**.

- The ***PGEName*** is the name of the PGE for which this static granule is being Inserted. For example, type **PGE1aT**, press **Enter**.
- The ***PGEName*** must match exactly the PGE name entered into the PDPS for this PGE (see Section 11.1.2).

12 At the program prompt **PGE version (default 1)?**, type ***PGEVersion***, press **Enter**.

- The ***PGEVersion*** is the version of the PGE entered in step 7 for which this static granule is being Inserted. For example, type **1**, press **Enter**.
- The ***PGEVersion*** must match exactly the PGE version entered into the PDPS for this PGE (see Section 11.1.2).

13 At the program prompt **Filename to insert?**, type ***GranuleFileName***, press **Enter**

- The ***GranuleFileName*** is the file name of the static data granule to be Inserted. For example, type **CADMERLW_1**, press **Enter**.

14 At the program prompt **Associated ASCII metadata (target MCF) filename to insert (default *GranuleFileName*.met)?**, press **Enter**.

- The default is the file name of the granule to insert with the .met file name extension. If the default is not correct, then the file name of this file must be entered.
- The static data granule will be Inserted to the IMF Data Server.

15 At the program prompt **Hit Enter to run again, 'q <Enter>' to quit:** type **q** and press **Enter** to quit or just press **Enter** to insert additional dynamic granules.

- If continuing, repeat steps 7 through 14.

Inserting Test Dynamic Files to the Data Server

This program is used to insert PGE input files to the Data Server. These files are "test dynamic", i.e., they are files that are normally inserted by Ingest in the Production system, but are test files here in the SSI&T environment. An example of such a file is a Level 0 test data file. Such test files are normally part of the DAP.

Before this program is used, the following must have occurred:

- An ESDT must be defined for the file. This is not a special "test ESDT", but is the same as the ESDT as used in the Production system.
- An file containing ASCII metadata for this ESDT granule must be constructed, in the format that the Data Server expects, i.e., a "target MCF".

- If it is desired to kick off a PGE through insertion of this file (simulating an insert by Ingest), then a subscription for the file must be registered just as in the Production system. This also requires normal DPR setup.

Creating a Target MCF for a Dynamic Granule

1 Log into the AIT 1 Sun platform by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun, type **cd *SourceMCFpathname***, press **Enter**.

- The ***SourceMCFpathname*** is the full path name of the directory containing the source MCF file for the dynamic data granule to be inserted into the IMF Data Server. For example, **cd /Rel_A/data/SourceMCF/**, press **Enter**.

4 At the UNIX prompt on the AIT Sun, type **SrcToTargetMCF**, press **Enter**

- This will invoke the source to target MCF template program and prompt the user for further information (see steps 5 - 9).

5 At the prompt **Enter source MCF file name:** type ***SourceMCFfilename***, press **Enter**

- The ***SourceMCFfilename*** is the file name of the source MCF for the data granule to be Inserted. For example, type **CER01T_1.MCF**, press **Enter**.

6 At the prompt **Enter target MCF filename:** type ***TargetMCFfilename.*met**, press **Enter**.

- The ***TargetMCFfilename*** is the file name of the target MCF template corresponding to the data granule to be Inserted. The **.met** file name extension is required for the target MCF file. For example, type **CER01T_1.met**, press **Enter**.

7 At the UNIX prompt on the AIT Sun, using a text editor to edit the target MCF file template. If *vi* text editor is used, type **vi *TargetMCFfileame.met***, press **Enter**.

- This command invokes the *vi* editor and reads in the ***TargetMCFfileame.met*** template file. For example, type **vi CER01T_1.met**, press **Enter**.

- For other text editors, refer to that editor's documentation.

8 Edit the target MCF template in accordance with the information provided from the science software (PGE). The following guidelines should be followed when performing the edition on the target MCF template file:

- For those objects with the **Data_Location="PGE"** in the source MCF, such as for those describe the date, time, and the file locations, the resulting data values for the corresponding target MCF file must be filled out by the SSIT operator. These data values should be obtained from the instrument team based on the specific PGE requirements.

9 Save the changes made to the target MCF template and exit the editor. Type **:wq**, press **Enter**.

- This will create a target MCF file for the dynamic data granule to be inputted into the IMS Data Server.

Inserting a Dynamic Data Granule into the IMF Data Server

---

1 Log into the AIT 1 Sun platform by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Change directories to the location of the SSIT Manager GUI, by typing the following UNIX command line in the XTerm window.

cd /usr/ecs/Rel_A/CUSTOM/bin/DPS then press the **Enter** key.

- The changed directory path is displayed in front of a new command line.

4 Invoke the SSIT Manager GUI by typing the following UNIX command line in the XTerm window.

DpAtMgr ConfigFile /usr/ecs/Rel_A/cfg/DpAtMG_GSFC.CFG & then press the **Enter** key.

- The SSIT Manager GUI is displayed.

5 At the UNIX prompt on the AIT Sun, type **cd *DataGranulePathname***, press **Enter**.

- The ***DataGranulePathname*** is the full path name of the directory containing the dynamic data granule(s) to Insert to the IMF Data Server.. For example, **cd /disk2/PGE32/data/**, press **Enter**.

6 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert Test Dynamic**.

- An xterm in which DpAtInsertTestFile.sh is running will be displayed.
- Alternatively, the same tool can be invoked by typing at a UNIX prompt on the AIT Sun **DpAtInsertTestFile.sh**, press **Enter**.
- The DpAtInsertTestFile.sh program will prompt for further information.

7 At the program prompt **Config filename (default *defaultConfigFile*)?**, press **Enter**.

- The default configuration file for this tool will be used.
- The ***defaultConfigFile*** will be replaced by the full path name and file name of the default configuration file. The file name will be DpAtID_*daac*.CFG where *daac* will be replaced by one of {GSFC, EDC, LARC, NSIDC}.

8 At the program prompt **ESDT name?** type ***ESDTShortName***, press **Enter**

- The ***ESDTShortName*** is the ShortName of the ESDT descriptor file corresponding to this granule to be Inserted. For example, type **CER01T**, press **Enter**.

9 At the program prompt **Filename to insert?** type ***GranuleFilename***, press **Enter**

- The ***GranuleFileName*** is the file name of the data granule to be Inserted. For example, type **CER01T_1**, press **Enter**.

10 At the program prompt **Associated ASCII metadata (target MCF) filename to insert (default *GranuleFileName*.met)?**, press **Enter**.

- The default is the file name of the granule to insert with the .met file name extension. If the default is not correct, then the file name of this file must be entered.
- The dynamic data granule will be Inserted to the IMF Data Server. For reference, the IMF Data Server Universal Reference (UR) will be printed on the screen.

11 At the program prompt **Hit Enter to run again, 'q <Enter>' to quit:** type **q** and press **Enter** to quit or just press **Enter** to insert additional dynamic granules.

- If continuing, repeat steps 7 through 10.

Insert EXE TAR

This program is used to insert the EXE TAR file (a.k.a the "SSEP", or Science Software Execution Package) to the Data Server. This tar file consists of all files needed to run a PGE, except for input data files. This includes the executables, any scripts, the SDP Toolkit runtime message files, and the Metadata Control File (MCF).

Before itÕs use, the following must have occurred:

- An ESDT must be defined for the file.
- An file containing ASCII metadata for this ESDT granule must be constructed, in the format that the Data Server expects, i.e., a "target MCF".
- The executables must have been compiled on the target science processor platform (currently SGI).
- The names of the top level shell (script or executable which kicks the PGE off) and the MCF files must have been entered on the PROFILE tab of the *PDPS/SSIT Database Operational Metadata Update GUI*.
- The files in question must all be put into a single tar file.

1 Log into the AIT 1 Sun platform by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun, type **mkdir *SSEPpathname***, press **Enter**.

- The ***SSEPpathname*** is the full path name of a *new* directory which will contain all the files to be placed into the SSEP as well as the SSEP itself.
- It is recommended that ***SSEPpathame*** be named with a convention that indicates the PGE for which a SSEP will be created. For example, type **mkdir MOD35.ssep**, press **Enter**.

4 At the UNIX prompt on the AIT Sun, type **cd *SSEPpathname***, press **Enter**.

- The ***SSEPpathname*** is the directory name of the new directory created in step 1.

5 At the UNIX prompt on the AIT Sun, type **cp *file1 file2 … filen* .**, press **Enter** (note the "dot" and then space at the end of the command).

- The ***file1, file2,…filen*** represents a list of file names (delimited by spaces) to copy into the current directory, ***SSEPpathame*** (the "dot" represents the current directory and must be last in the command).
- For example, type **cp MOD35.pge mod35.mcf MOD_13453 .**, press **Enter** (note the space and then "dot" at the end of the command).
- The files copied into this directory should be the PGE executable, any shell scripts or other executables that are part of the PGE, SDP Toolkit message files, and the Bourne shell profile (if applicable).
- Files can be individually copied into the ***SSEPpathame*** directory. For example, type **cp MOD35.pge .**, press **Enter** (note the space and then "dot" at the end of the command). Repeat for each file needed in the SSEP for this PGE.

6 At the UNIX prompt on the AIT Sun, type **tar cvf *SSEPfilename*.tar ***, press **Enter**.

- The ***SSEPfilename*.tar** is the file name for the SSEP tar file. The file name extension .tar is recommended but not required.
- The asterisk (*****) is a file name wildcard that represents all files in the current directory. This will place all files in the SSEP tar file.
- Once created, the contents of the SSEP tar file can be viewed by typing **tar tvf *SSEPfilename*.tar**, press **Enter**.

7 At the UNIX prompt on the AIT Sun, type **cp *filenme.met.tpl filename.met,*** press **Enter**.

- The ***filename.met.tpl*** is the file name of the template Target MCF for this SSEP.
- The ***filename.met*** is the file name of the Target MCF to be tailored for this SSEP.

8 At the UNIX prompt on the AIT Sun, type **vi *filename.met***, press **Enter**.

- The ***filename.met*** is the Target MCF for this SSEP.
- This command invokes the *vi* editor. Edit the ***filename*.met** with the specific information for the SSEP to be inserted.
- The following guidelines should be followed when editing on the Target MCF (***filename*.met**):
  - The value for the VERSIONID object should be filled out with the proper PGE version. For example: "1" .
  - In the INFORMATIONCONTENTCONTAINER object,
    - The value for the PARAMETERNAME object of the class "1" should be filled out with the PGE name. For example: "BTS".
    - The value for the PARAMETERNAME object of the class "2" should be filled out with the PGE Science Software Version. For example: "1".
    - The value for the PARAMETERNAME object of the class "3" should be filled out with the Platform Name. For example: "IRIX".
    - The value for the PARAMETERNAME object of the class "4" should be filled out with the Platform Version. For example: "6.2".
    - The value for the PARAMETERNAME object of the class "5" should be filled out with the date to perform the Insertion. For example: "970319".
    - The value for the PARAMETERNAME object of the class "6" should be filled out with the time to perform the Insertion. For example: "14:45:00".

9 Save the changes made to the SSEP's Target MCF (***filename*.met**) and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Enter**.
- For other editors, refer to that editor's documentation.

Inserting a Science Software Executable Package to the IMF Data Server

1 Log into the AIT 1 Sun platform by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun, type **cd *PGEpathname***, press **Enter**.

- The ***PGEpathname*** is the full path name of the directory containing the SSEP to Insert to the IMF Data Server (see Section 11.4.1 for assembling a SSEP). For example, **cd /disk2/PGE32/bin/**, press **Enter**.

4 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert EXE TAR**.

- An xterm in which DpAtInsertExeTarFile.sh is running will be displayed.
- Alternatively, the same tool can be invoked by typing at a UNIX prompt on the AIT Sun **DpAtInsertExeTarFile.sh**, press **Enter**.
- The DpAtInsertExeTarFile.sh program will prompt for further information.

5 At the program prompt **Config filename (default *defaultConfigFile*)?**, press **Enter**.

- The default configuration file for this tool will be used.
- The ***defaultConfigFile*** will be replaced by the full path name and file name of the default configuration file. The file name will be DpAtIE_*daac*.CFG where *daac* will be replaced by one of {GSFC, EDC, LARC, NSIDC}.

6 At the program prompt **mode (default ops)?**, press **Enter**

- If **ops** is not the correct mode, then enter in the correct mode (*e.g.* SSIT) and press **Enter**.

7 At the program prompt **PGE name?**, type ***PGEName***, press **Enter**.

- The ***PGEName*** is the name of the PGE for which this static granule is being Inserted. For example, type **PGE1aT**, press **Enter**.
- The ***PGEName*** must match exactly the PGE name entered into the PDPS for this PGE (see Section 11.1.2).

8 At the program prompt **Science software version (default 1)?**, type ***SSWversion***, press **Enter**.

- The ***SSWversion*** is the version of the science software which is being Inserted in this SSEP. Press **Enter** to accept the default or enter in a version and press **Enter**.

9 At the program prompt **Filename to insert?**, type ***SSEPFileName***, press **Enter**

- The ***SSEPFileName*** is the file name of the SSEP to be Inserted. For example, type **MOD35_1**, press **Enter**.

10 At the program prompt **Associated ASCII metadata (target MCF) filename to insert (default *SSEPFileName*.met)?**, press **Enter**.

- The default is the file name of the granule to insert with the .met file name extension. If the default is not correct, then the file name of this file must be entered.

11 At the program prompt **Top level shell filename within the tar file?**, type ***ExecFileName***, press **Enter**.

- The ***ExecFileName*** is the file name of the top level executable within the SSEP tar file. It should be the same as was entered into the PDPS/SSIT Database Update GUI (Section 11.1.3).

- The SSEP will be Inserted to the IMF Data Server.

12 At the program prompt **Hit Enter to run again, 'q <Enter>' to quit:** type **q** and press **Enter** to quit or just press **Enter** to insert additional dynamic granules.

- If continuing, repeat steps 5 through 12.

Linking a PGE with DAAC Version of SDP Toolkit

The compiling and linking of science software will vary according to the particular delivery. This procedure will follow some of the "typical" steps that may or may not be applicable to a particular situation. Most of the following steps will be unnecessary if the science software has been previously built successfully with the SCF version of the SDP Toolkit.

The following is a list of tools, and or assumptions:

- The science software was previously built successfully using the SCF version of the SDP Toolkit.

- All instructional information supplied with the delivery should be read prior to compiling and linking a PGE with the SCF version of the SDP Toolkit. These instructions should be the primary source of information.

- The science software delivery has been unpacked and placed into the software build area.

- All files necessary to build the science software are available, accessible, and have the proper permissions set.

- The build process will be done on the SGI Power Challenge in a UNIX shell.

Linking a PGE with DAAC Version of SDP Toolkit

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Change directories to the location of the SSIT Manager GUI, by typing the following UNIX command line in the XTerm window.

cd /usr/ecs/Rel_A/CUSTOM/bin/DPS then press the **Enter** key.

- The changed directory path is displayed in front of a new command line.

4 Invoke the SSIT Manager GUI by typing the following UNIX command line in the XTerm window.

DpAtMgr ConfigFile /usr/ecs/Rel_A/cfg/DpAtMG_GSFC.CFG & then press the **Enter** key.

- The SSIT Manager GUI is displayed.

5 Select the **Tools→Xterm** from the SSIT Manager menu bar.

- An Xterm window is displayed.

6 Setup the toolkit environment by typing**:**

seten PGSHOME toolkithomedir (where **toolkithomedir** is the home directory of the desired toolkit) then press **Enter**.

source $PGSHOME/bin/sgiX/pg s-dev-env.csh (in sgiX refers to the appropriate processor) then press **Enter**.

7 Check out the make file from ClearCase™ by typing:

cleartool setview viewname (where **viewname** is the name of the view created for this PGE) then press **Enter**.

cd path (where **cd path** is the name of the directory in the VOB where the make file was checked in) then press **Enter**.

cleartool checkout makefile -nc (where **makefile** is the name of the make file to be used with this PGE) then press **Enter**.

8 Examine and alter any supplied make files by typing:

vi makefile (where **makefile** is the name of one of the make files then press **Enter**.

- Refer to supplied documentation.
- Check that the compiler, compiler flag settings and other environment variable settings are appropriate.

$(ADD_LFLAGS) then press **Enter**.

$(ADD_LIBS) then press **Enter**.

- Try to make use of environment variables pre-set used by the shell rather than setting them within the make/build files.

9 Compile any SMF files by typing: **smfcompile -f textfile.t** (where **textfile.t** is the filename of the status message text file delivered with the science software) then press **Enter**.

- Only one textfile at a time can be processed.
- -r (automatically copies runtime files to the directory) and -i (automatically copies include files to the directory) options can be used.
- While this step may have been performed in compiling the software with the SCF Toolkit, part of it still needs to be performed for the DAAC Toolkit.

10 Go to the directory containing the status message files by typing: **cd pathname** (where **pathname** is the directory containing the SMF files) then press **Enter**.

- SMF files have the extension .t attached to the end of the line (example: MODIS_39501.t.

11 Place the created include files and the created runtime files into the proper directories by typing:

mv includefilename $PGSINC (where **includefilename** is the name of the newly created include file (PGS_INSTR_XXXX.h)

mv runtimefilename $PGSMSG (where **runtimefilename** is the name of the newly created runtime file (PGS_XXXX.h). XXXX is the seed number that was in the corresponding .t file.

12 Repeat steps 7 and 8 as necessary for each of the SMF files associated with the science software.

- Only one textfile at a time can be processed. Wildcards cannot be used.

13 Verify that the directory structure matches the directories expected by the build script.

- This step will be unnecessary if the software had been previously built successfully with the SCF version of the SDP Toolkit.
- Deliveries may come with install scripts which place files into various directories according to some structure.

14 Checkout all executable or object files for this PGE from ClearCase™ by typing: **cleartool checkout -nc filename** (where **filename** is the name of the executable or object file to be checked out) then press **Enter**.

- It is not recommended that executables and object files be checked into ClearCase™.

15 Once installation scripts, make files and/or build scripts are setup properly, perform the build according to the instructions delivered with the PGE.

- Science software deliveries may come with a single, top-level script to do the entire build or the build process could involve a series of steps, each of which should be described fully in the delivered documentation.

16 Check in the make file from ClearCase™ by typing:

cleartool checkin makefile -nc (where **makefile** is the name of the make file be used with this PGE) then press **Enter**.

This page intentionally left blank.

PGE Planning and Processing

During Science Data Production

After a PGE has completed SSI&T at the DAAC, the PGE is production ready and will reside on the Production Data Server. The PDPS database will be populated with the following information:

- Inputs required by the PGE.
- Outputs produced by the PGE.
- Resource requirements.
- Execution conditions.

Planning

The ECS Planning Subsystem provides the DAAC operations staff with the ability to create, modify, and implement production plans. Production plans are based on combining PGE information contained in the PDPS database with local DAAC strategies for fulfilling production requests

Production Request (PR) describes an order for data that is to be produced. There are three different types of PRs.

- Routine - a request that signifies the processing of new data.
- Reprocessing - a request that signifies the reprocessing of data.
- On-demand - a request that signifies a request for an unplanned generation of a particular data product.

A production plan is comprised of multiple production requests. A production request is built from one or more data processing requests.

A data processing request (DPR) describes the run of a PGE including the specific input and output data, filenames, run-time parameters, dependencies and predicted run-times. One or more DPRs may be required to satisfy a production request.

Creating a New Production Request

Creating a new PR is performed by the Production Planner. The Production Planner uses the Production Request Editor GUI from the Production Planner desktop. The Production Planner will require the following information:

- Name of the PR.
- Priority of the PR.
- PGE to be used in processing the PR.
- PGE parameters to be used.
- Parameter(s) to be changed (optional).
- Start Date.
- Start Time.
- End Date.
- End Time.
- Comments (if applicable).

Creating a New Production Request

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

- If you access the workstation through a remote login (rlogin), you must enter **xhost +** and then press the **Enter** key.
- Once you have entered xhost +, but prior to the remote login, enter **setenv DISPLAY** <local workstation IP address>**:0.0** where the local workstation IP address represents the IP address you where you are located**.**
- You may need to setup the terminal so that the remote host is displayed on your screen. (Sun machine) This is done by clicking on the **Application Manager** icon (the file drawer located at the bottom of the screen), followed by the **Desktop Tools** icon**,** followed by the **Terminal Consol** icon, then typing **xhost + plnn1sun** or **plnn1sun.gsfc.ecs.nasa.gov** in the display window.

3 Perform a remote login by typing **rlogin [plnn1sun]** then press **Enter.**

- The **Enter Password** prompt is displayed.

4 Enter the **password** then press **Enter.**

5 Enter the directory where the setup script is located by typing **cd [**directory name**]** then press **Enter**.

6 Source the setup script by typing **source [**script name ex. PR_Setup.com**]** then press **Enter.**

7 Enter the script alias by typing **[**alias name ex. PREditor**]** then press **Enter**.

- The Production Request Editor GUI is displayed.

8 Select the Production Request Editor by clicking (single click) on the **PR Edit** tab.

- The **PR Edit** GUI page is (Figure 8) presented.

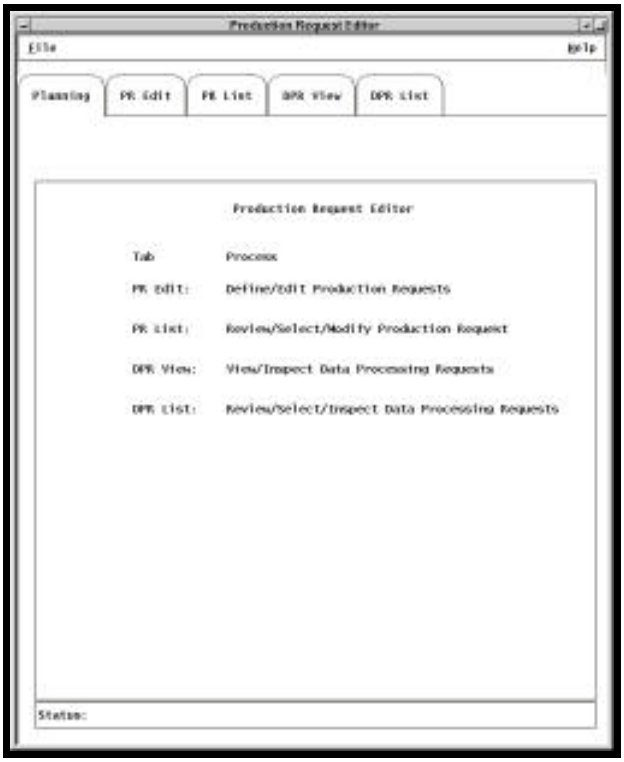

Figure 7. Production Request Editor Introductory GUI

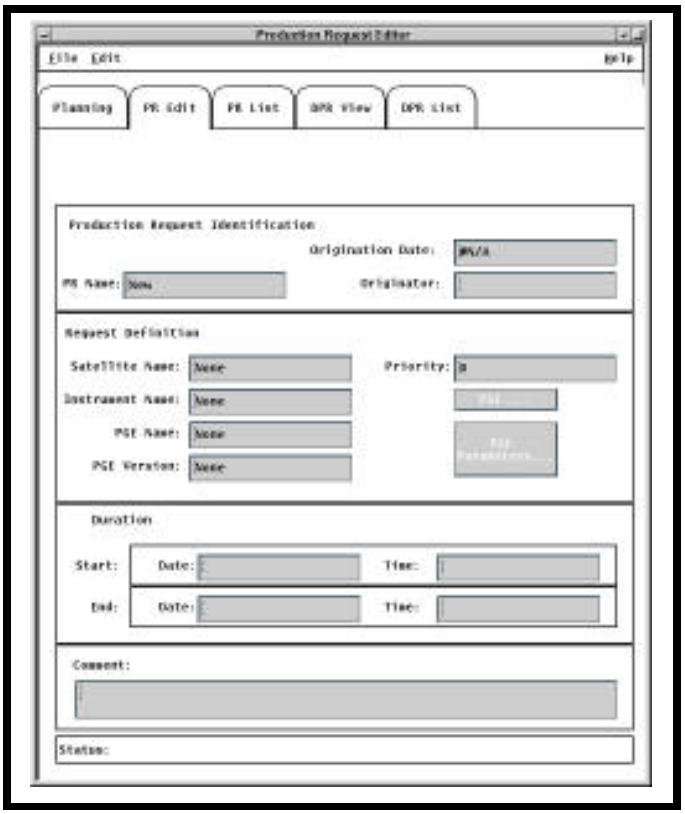

Figure 8. PR Edit GUI

9 Type the name of the PR in the **PR Name** field.

- The **Origination Date** and **Originator** fields will be automatically filled when the Production Request is saved at the end of the procedure. (You do not fill in these fields.)

10 Type the priority for the PR in the **Priority** field.

- Remember that the job **Priority** field specifies the priority of the job with 1 (one) as the highest priority and 99 the lowest priority.

11 Click on the **PGE...** button.

- The **PGE Selection** GUI is displayed.

12 Select the desired PGE from the list by clicking on the appropriate row in the table.

- The PGE list is scrollable. (If there are items on the list in addition to those currently visible in the window, the additional items can be viewed by clicking on the arrows associated with the scrollbars.)

13 Click on the **OK** button.

- The **Production Request** - **PR Edit** GUI page (Figure 8) is displayed.

The following fields are automatically filled:

— **Satellite Name**

— **Instrument Name**

— **PGE Name**

— **PGE Version**

14 Click on the **PGE Parameters...** button.

- The **PGE Parameter Mappings** GUI is displayed.
- The **PGE Parameter Mappings** GUI has a table that lists the following information:

— **Parameter Name**.

— **Logical Id**.

— **Default Value**.

— **Override Value**.

— **Description.**

15 If any PGE parameter(s) should be changed, select (highlight) a parameter to be changed by clicking on the corresponding row in the list of parameters.

- The parameter row is highlighted.
- If you do not change the parameters, the values in the Default column will be used.
- The PGE parameter mappings list is scrollable.

16 Type the desired value in the **Parameter Mapping** field.

- The value in the **Override Value** column is updated.
- If any other parameter is to be changed, highlight it and type the desired value in the Parameter Mapping field.

17 Click on the **OK** button to approve the changes.

- The **Production Request** - **PR Edit** GUI is displayed.

18 Type the desired start date (in ***MM/DD/YY*** format) in the **Duration — Start — Date** field, then press the **Tab** key on the keyboard to advance to the next field.

19 Type the desired start time (in ***hh:mm*** format) in the **Duration — Start — Time** field, then press the **Tab** key on the keyboard to advance to the next field.

20 Type the desired end date (in ***MM/DD/YY*** format) in the **Duration — End — Date** field, then press the **Tab** key on the keyboard to advance to the next field.

21 Type the desired end time (in ***hh:mm*** format) in the **Duration — End — Time** field, then press the **Tab** key on the keyboard to advance to the next field.

22 Type any relevant comments in the **Comments** field.

23 Select **Save** from the **File** pull-down menu (**File** → **Save**).to save the production request.

- The Production Request is saved.
- The **Origination Date** and **Originator** fields are automatically updated.
- If no PR name has been entered in the **PR Name** field, you will be prompted to enter a PR name during the “Save” process.

Creating and Activating a Production Plan

The Production Planner creates a plan for production data processing at the DAAC by selecting specific PRs whose DPRs are to be run. The planning tool provides a forecast of the start and completion times of the jobs based upon historical experience in running these PGEs. Through the planning tool, when the generated plan is "activated," the information included in the plan is transferred to the Data Processing subsystem and loaded into the Platinum AutoSys tool where production processing is managed.

The Production Planner creates the plan by selecting PRs from two lists of PRs, i.e., the list of available "Unscheduled" PRs and the list of "Scheduled" PRs. Using arrow buttons, the Production Planner moves the PRs between lists until the "Scheduled" list contains the desired set of PRs that define the new plan.

Before creating a new production plan the Production Planner must have available the following information:

- Name of the plan.
- Comments (if any).
- PRs to be included in the new production plan.

Creating a New Production Plan

---

1 Log into the workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

- If you access the workstation through a remote login (rlogin), you must enter **xhost +** and then press the **Enter** key.
- Once you have entered xhost +, but prior to the remote login, enter **setenv DISPLAY <**local workstation IP address>**:0.0** where the local workstation IP address represents the IP address you where you are located**.**
- You may need to setup the terminal so that the remote host is displayed on your screen. (Sun machine) This is done by clicking on the **Application Manager** icon (the file drawer located at the bottom of the screen), followed by the **Desktop Tools** icon**,** followed by the **Terminal Consol** icon, then typing **xhost + plnn1sun** or **plnn1sun.gsfc.ecs.nasa.gov** in the display window.

3 Perform a remote login by typing **rlogin [plnn1sun]** then press **Enter.**

- The **Enter Password** prompt is displayed.

4 Enter the **password** then press **Enter.**

5 Enter the directory where the setup script is located by typing **cd [**directory name**]** then press **Enter**.

6 Source the setup script by typing **source [**script name ex. PLWB_Setup.com**]** then press **Enter.**

7 Enter the script alias by typing **[**alias name ex. PREditor**]** then press **Enter**.

- **Planning Workbench** GUI is displayed (Figure 9).

Figure 9. Planning Workbench GUI

- Data concerning the currently active production plan are displayed.
- If you want to "kill" (deactivate) the currently active production plan without activating a replacement, click on the **Kill** button.
- Whenever you activate a plan (by clicking on the **Activate** button), you automatically "kill" the currently active plan.

8 Select **File** → **New** from the pull-down menu.

- The "New" window appears.

9 Type a name for the new plan, then press the **Tab** key on the keyboard.

- The **Planning Workbench** GUI is displayed.
- The **Plan Name** is displayed.
- The **Status** displayed is **Candidate**.

10 Type the desired date (in ***MM/DD/YY*** format) in the [TBD], then press the **Tab** key on the keyboard to advance to the next field.

11 Type the desired time (in ***hh:mm*** format) in the [TBD], then press the **Tab** key on the keyboard.

- The **Rollover Time** is displayed.

12 Type any relevant comments (up to 255 characters) in the **Comments** field.

13 Move PRs between the **Unscheduled** and **Scheduled** lists as necessary by selecting (highlighting) the PR to be moved by clicking on the PR in the list from which it is to be moved then clicking on the up or down arrow button (as applicable) to move the PR to the other list.

- Highlighted PR disappears from one list and appears on the other.
- The unscheduled and scheduled PR lists are scrollable.

14 When the **Scheduled** list accurately reflects the PRs to be scheduled in the production plan, select **File** → **Save** (or **File** → **Save As**) from the pull-down menu to save the new production plan.

- The new production plan is saved.

15 If the new plan is to be activated immediately, click on the **Activate** button to activate the new plan.

- The currently active plan is killed (deactivated) and the new plan is activated.
- The **Production Planning Timeline** GUI is displayed.

16 If the new production plan is to be used as a baseline plan, click on the **Baseline** button.

- The "New" window appears.
- The plan is recorded as well as the time of baselining so that it can be used in comparing future processing results with planned objectives.

17 If the production plan being displayed is active and should be deactivated, click on the **Kill** button.

- The "New" window appears.
- The plan is deactivated without activating another plan.

Processing

Once a candidate plan has been activated, each the DPRs will result in subscriptions to the Data Server for the data needed. A request will go to the Data Server asking for notification when the required input data arrives.

Planning knows what data to request from the Data Server because the PDPS database stores this information as determined by the ESDT for each PGE. When the Data Server receives new data, it routinely checks to see if there are any outstanding subscriptions. If there are subscriptions, Planning will be notified. Once the input data required by a DPR becomes available, the DPR can be queued for processing.

Staging - The Data Processing Subsystem requests that the required input data, PGE (binary executables and shell scripts) and SDP Toolkit files be placed on a disk set aside for processing.

Process Control File (PCF) - establishes a linkage between logical Ids that the science software uses and the physical files that exist on the staging disk.

After the PGE has completed, the DPS will deallocate resources.

A Production History file will be created and will contain information concerning the conditions that the data products were generated by the PGE.

Post-Processing Activities

Viewing Product-Created Metadata Using EOS View Tool

Another feature of the SSIT Manager is the EOS View Tool. This tool allow a user to check metadata of a data product created by running a PGE.

The following is a list of tools, and or assumptions:

- The SSIT Manager is running on a SUN workstation.
- The science software being used to generate a data product has been ingested, built and successfully run in the Testbed environment.
- The product created has been written out as a HDF File and contains metadata.

View Metadata using EOS View Tool

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start up EOS View to view product metadata by selecting **Tools→Product Examination→EOSView** menu from the SSIT Manager GUI.

- The EOS View GUI will be displayed.

4 Open a HDF product file from which metadata is to be viewed by selecting the **File→ Open** button from the main menu bar.

- A **File Selection Dialog** window will open and the user should be able to select the appropriate directory and file to open.
- Once the desired product file has been opened, the specific types of HDF objects in the file will be listed in the **Contents** window.

5 From the **Contents** window double-click on a particular HDF Object (Vgroup, SDS, etc.) to get to the metadata.

- Metadata is refereed to as **attribute data.**
- The structure of the HDF object will appear in a dialog window with buttons on the bottom portion of the window to view the data of the object itself.

6 Display the attribute (i.e. metadata) of this particular HDF object by selecting the **Table** button to display the table data of the object.

7 View the attribute (i.e. metadata) of this particular HDF object by selecting the **File-Attribute** button.

- Any metadata associated with the object will be displayed in another text window.

8 Quit when done by typing Q then press **Enter.**

Viewing Granule and Data Dictionary Metadata

The SSIT Manager can use the EOS View Tool to check entries in the Metadata Control File (MCF) for each PGE generating a data product and checks the MCF data for valid range values with the Data Dictionary. This tool allows a user to check the metadata of a data product created by running a PGE.

The following is a list of tools, and or assumptions:

- The SSIT Manager is running on a SUN workstation.
- The science software being used to generate a data product has been ingested, built and successfully run in the Testbed environment.
- Each PGE has been checked for mandatory tool compliance.
- A text editor is available to the SSI&T personnel

View Granule and Data Dictionary Metadata using EOS View Tool

---

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 Start up EOS View to view product metadata by selecting **Tools→Product Examination→EOSView** menu from the SSIT Manager GUI.

- The EOS View GUI will be displayed.

4 Open a Metadata Configuration File (MCF) or Data Dictionary (DD) by selecting the **File Open** button from the main menu bar.

- A **File Selection Dialog** window will open and the user should be able to select the appropriate directory and file to open.
- Once the desired file has been opened, the specific types of MCF or DD objects in the file will be listed in the **Contents** window.

5 From the **Contents** window double-click on a particular MCF or DD Object (Vgroup, SDS, etc.) to get to the metadata.

- Metadata is referred to as **attribute data.**
- The structure of the MCF or DD object will appear in a dialog window with buttons on the bottom portion of the window to view the data of the object itself.

6 Display the attribute (i.e. metadata) of this particular MCF or DD object by selecting the **Table** button to display the table data of the object.

7 View the attribute (i.e. metadata) of this particular MCF or DD object by selecting the **File-Attribute** button.

- Any metadata associated with the object will be displayed in another text window.

8 Use any desired text editor (vi, emacs, xedit, etc.) to view these files.

- This step allows the contents of the MCF or DD to be checked for such things as mandatory metadata.

9 Correct the MCF or DD if errors such as ODL syntax of range of values listed in MCF or DD are invalid.

10 Quit when done by typing Q then press **Enter.**

Troubleshooting and General Investigation

Examining PGE-Produced Log Files

When the PGE is run outside of the PDPS, the PCF specifies the location and file names of the log files produced. This procedure describes how to locate that information from the PCF and use it to examine the log files.

The following is a list of tools, and or assumptions:

- The Data Processing Request ID (DPR ID) for PGE execution is required for this procedure.
- The SDP Toolkit writes the Log Status, Log User, and Log Report files to the directories specified in the PCF.
- All required environmental variables have been set.
- The operations/test account has read access to the PDPS Database.
- The .cshrc file should have the following entries.
    - setenv SYBASE/vendor/sybase
    - setenv SYROOT $SYBASE/sybooks
    - setenv EBTRC $SYBROOT/sun5m/.ebtrc
    - setenv DSQUERY computer-server (where computer-server is system dependent)
    - set path = ($path $SYBASE $SYBASE/bin $SYBROOT/sun5m/bin
- A text editor is available to the SSI&T personnel

LogStatus captures all error and status information concerning a program.

LogUser captures a subset of messages of level or type "_U_" or "_N_" which are of particular interest to a user.

LogReport captures arbitrary message strings sent by the PGE software. The messages are unrelated to the toolkit functions, but will contain important messages for scientists and developers.

Examining PGE-Produced Log Files for PGEs Run Outside of PDPS

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **cd *PCFpathname***, press **Enter**.

- The ***PCFpathname*** is the full path name to the location of the PCF used by the PGE for which log files are to be examined.

4 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *PCFfilename***, press **Enter**.

- The ***PCFfilename*** is the file name of the PCF used by the PGE for which log files are to be examined.
- This brings up the file named ***PCFfilename*** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs MOD35.pcf**, press **Enter**.

5 In the editor, search for logical IDs (beginning in the first column) **10100**, **10101**, and **10102**. These are the PCF entries for the Status log, User log, and Report log respectively. For each, note the file names in field 2 and the path names in field 3. Then quit the editor.

- If field 3 is blank, then the location is given by the default location specified in a line above the entries beginning with the "!" character.

6 At the UNIX prompt on the SDPS SGI, type **vi *StatusLogPathname/filename***, press **Enter**.

- The ***StatusLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10100. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs /PGE/MOD35/LogStatus**, press **Enter**.

7 At the UNIX prompt on the SDPS SGI, type **vi *UserLogPathname/filename***, press **Enter**.

- The ***UserLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10101. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs /PGE/MOD35/LogUser**, press **Enter**.

8 At the UNIX prompt on the SDPS SGI, type **vi *ReportLogPathname/filename***, press **Enter**.

- The ***ReportLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10102. When finished, quit the editor.
- Note any anomalous messages in file.
- Any text editor may be used such as *emacs*. For example, **/PGE/MOD35/LogReport**, press **Enter**.

Production History Log Files From PGEs Run Within the PDPS

The Production History (PH) is created during PGE execution within the PDPS and then Inserted into the IMF Data Server upon PGE completion. Included in the PH are the PGE log files. To access a Production History associated with a particular PGE run requires the DPR ID of the PGE run and the full path name to the IMF Data Server Production History archive.

---

1 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **cd *PHArchivePathname***, press **Enter**.

- The ***PHArchivePathname*** is the full path name of the IMF Data Server Production History archive.
- For example, type **cd /net/sprg1sgi/imf_data/archive/PH**, press **Enter**.

2 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **ls -al**, press **Enter**.

- A list of the current contents will be displayed. These will be Production History tar files. The names of these files consist of the DPR ID, the collection time, and the insertion time.
- Look for the PH corresponding to the DPR ID of interest.

3 At a UNIX prompt on the AIT Sun or on the SDPS SGI, type **cp *PHtarFilename WorkingPathname***, press **Enter**.

- The ***PHtarFilename*** is the file name of the Production History tar file.
- The ***WorkingPathname*** is the full path name to some working directory in which the Production History tar file is to be placed and examined.

4 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **cd *WorkingPathname***, press **Enter**.

- The ***WorkingPathname*** is the full path name to the working directory specified in step 3.

5 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **tar xvf *PHtarFilename***, press **Enter**.

- The ***PHtarFilename*** is the file name of the Production History tar file in the working directory.
- This command will untar the Production History tar file, extracting its component files into the current directory.

6 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *StatusLogFilename***, press **Enter**.

- The ***StatusLogFilename*** is the file name of the Status log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs LogStatus**, press **Enter**.

7 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *UserLogFilename***, press **Enter**.

- The ***UserLogFilename*** is the file name of the User log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs LogUser**, press **Enter**.

8 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *ReportLogFilename***, press **Enter**.

- The ***ReportLogFilename*** is the file name of the Report log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs LogReport**, press **Enter**.

Examining PDPS-Related Scripts and Message Files

This section describes how users may access files, in addition to the PGE-produced log files, which are created during the execution of a DPR job and which may hold information useful in tracing processing problems.

Some of these files are written by default to directory paths that can only be accessed on either the SGI processor machine or one of the Sun workstations. More detailed descriptions of these files and the conditions under which they are generated will be supplied in future Green Book versions.

Examining AutoSys JIL Scripts

JILxxxxxxxxx is the Job Information Language (JIL) script that defines the DPR job to **AutoSys** and which must be submitted to the **AutoSys** Database before a DPR job can be run. The name of the file created is system-generated and begins with the characters 'JIL' followed by nine characters (e.g. JILAAAa0066c).

Examining JILxxxxxxxx scripts on the AIT Sun

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun, type **cd *JILscriptPathname***, press **Enter**.

- The ***JILscriptPathname*** is the full path name to the location of the JILxxxxxxxx scripts to be examined.

4 At the UNIX prompt on the AIT Sun, type **vi *JILscriptFilename***, press **Enter**.

- The ***JILscriptFilename*** is the file name of the JILxxxxxxxx script to be examined.
- This brings up the file named ***JILscriptFilename*** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs *JILscriptFilename***, press **Enter**.

Examining the DpAtExecutionMain.out and DpAtExecutionMain.err Files on the SDPS SGI

DpAtExecutionMain.out and DpAtExecutionMain.err contain the PDPS-level error and status messages generated during the execution of a DPR job. These two files can be written to a directory on a SGI disk volume when a DPR job is executed. DpAtExecutionMain.out contains stdout messages resulting from the PGE itself; these may be generated from a user script which executes the PGE and may contain different information from that in the Log* files. DpAtExecutionMain.err contains stderr messages resulting from the execution of the PDPS (i.e. the executable DpAtExecutionMain.) The user can ensure that the files are generated by setting the environment variables DPAT_STD_ERR and DPAT_STD_OUT. The location and names of these files can be user-defined by setting the environment variables DPAT_STD_ERR and DPAT_STD_OUT. Setting these variables locally ensures that only a user's own DPR job messages are written to these files. DPAT_STD_ERR and DPAT_STD_OUT are not set automatically and the files will not be created automatically. If these files are desired, the user must explicitly set DPAT_STD_ERR and DPAT_STD_OUT prior to bringing up the SSIT Manager and executing the PDPS prototype software and in a manner such as:

setenv DPAT_STD_ERR /user/defined/directory/filename1

setenv DPAT_STD_OUT /user/defined/directory/filename2

Examining DpAtExecutionMain.out and DpAtExecutionMain.err on the SDPS SGI

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the SDPS SGI, type **cd *DpAtExecutionMain.outPathname***, press **Enter**.

- The ***DpAtExecutionMain.outPathname*** is the full path name to the location of the DpAtExecutionMain.out file to be examined.

4 At the UNIX prompt on the SDPS SGI, type **vi DpAtExecutionMain.out**, press **Enter**.

- This brings up the file named **DpAtExecutionMain.out** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs DpAtExecutionMain.out**, press **Enter**.

5 At the UNIX prompt on the SDPS SGI, type **cd *DpAtExecutionMain.errPathname***, press **Enter**.

- The ***DpAtExecutionMain.errPathname*** is the full path name to the location of the DpAtExecutionMain.err file to be examined.

6 At the UNIX prompt on the SDPS SGI, type **vi DpAtExecutionMain.err**, press **Enter**.

- This brings up the file named **DpAtExecutionMain.err** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs DpAtExecutionMain.err**, press **Enter**.

Examining the Event Logger Log File

The eventlogger.log file is indicated by the following science software PCF entry in the SUPPORT OUTPUT FILES section:

```
10113|eventlogger.log||||1
```

Because this file is specified in the science software PCF, the user can define both the directory path and the file name.

Examining the eventLogger Log File

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **cd *PCFpathname***, press **Enter**.

- The ***PCFpathname*** is the full path name to the location of the PCF used by the PGE for which the eventLogger.log file is to be examined.

4 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *PCFfilename***, press **Enter**.

- The ***PCFfilename*** is the file name of the PCF used by the PGE for which log files are to be examined.
- This brings up the file named ***PCFfilename*** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs *PCFfilename*** , press **Enter**.

5 In the editor, search for logical ID (beginning in the first column) **10113**. This is the PCF entry for the eventLogger.log file. Note the path name in field 3. Then quit the editor.

- If field 3 is blank, then the location is given by the default location specified in a line above the entries beginning with the "!" character.
- Note the file and path names.

6 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi *eventLogger.logDir/eventLogger.log***, press **Enter**.

- The ***eventLogger.logDir/eventLogger.log*** is the full path name and file name of the eventLogger.log file noted in step 3 associated with PCF logical ID 10113. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs *eventLogger.logDir/eventLogger.log***, press **Enter**.

Examining the PDPS Event Log File

This file is pointed to by the environment variable DPAT_EVENTLOG and is written to as DpAtExecutionMain executes.

Examining the PDPS Event Log

---

1 Log into the aitn1sun or aitn2sun workstation by typing: **username** then press **Enter**.

2 Enter the **password** then press **Enter.**

3 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **cd *pdps_event.logPathname***, press **Enter**.

- The ***pdps_event.logPathname*** is the full path name to the location of the PDPS event log file to be examined.

4 At the UNIX prompt on the AIT Sun or on the SDPS SGI, type **vi pdps_event.log**, press **Enter**.

- This brings up the file named **pdps_event.log** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs pdps_event.log,** press **Enter**.

Practical Exercise

Introduction

This exercise is designed to practice the key features of the SS I&T lesson.

Equipment and Materials

One ECS planning workstation.

Sample Science Software.

Science Software Integration and Test -

Situation: A new PGE and associated files has been developed by the Instrument Team. This exercise will require the CM Administrator and the Science Data Coordinator to prepare the PGE for installation and operation on the Production Server.

1. Acquire the science software and provide these files to the CM Administrator.
2. Import these file into ClearCase™. This may require creation of a view and a VOB to place these files.
3. Perform the initial setup of the SSIT Manager and verify that the SSIT Manager is prepared to use the SDP Toolkit, PCF files, and that all required environmental variables point to locations in the SDP Toolkit directory.
4. Verify that the code is in compliance with the ESDIS Standards.
5. Invoke the Prohibited Function Checker on the source files.
6. Verify that the PCFs are syntactically correct and contain all necessary information for the PGEs to run within the ECS DAAC production environment.
7. Resolve any conflicts found when checking the PCFs.
8. Compile the SMF or Error Status Message.
9. Compile the PGE and Link the with the SDP Toolkit.
10. Run the PGE (as an engineering Version).
11. Perform an HDF file comparison.
12. Examine the PGE LogStatus and LogReport.

This page intentionally left blank.

Slide Presentation

Slide Presentation Description

The following slide presentation represents the slides used by the instructor during the conduct of this lesson.